PATENT ABSTRACTS OF JAPAN

(11)Publication number : 2001-312324

(43)Date of publication of application : 09.11.2001

(51)Int.Cl. G06F 1/00 G06F 13/00
G06F 15/00 G06F 17/60

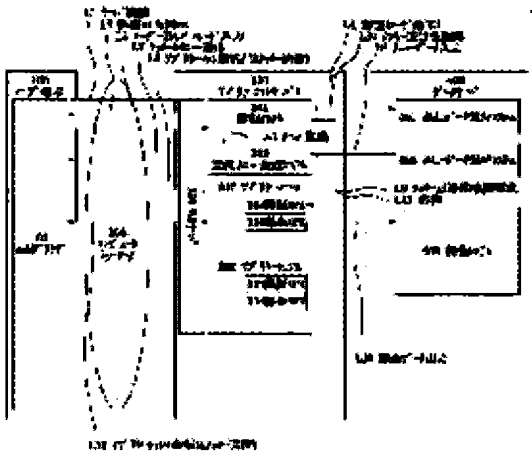
(21)Application number : 2000-131796

(22)Date of filing : 28.04.2000

(71)Applicant : NEC NEXSOLUTIONS LTD

(72)Inventor : MORITA MASAO

(54) METHOD FOR AUTHENTICATING AND CHARGING USER, RECORDING MEDIUM THEREFOR, METHOD AND SYSTEM FOR PROVIDING APPLICATION SERVICE



(57)Abstract:

PROBLEM TO BE SOLVED: To provide a method and a system for providing an application service, with which a vender can be easily participated in the business of an application service provider.

SOLUTION: A previously authenticated certificate is inputted to an application program(AP), the previously authenticated certificate is inputted through an authenticating API to an authenticating system, a certificate authenticated again is outputted through the authenticating API to the AP, a conversation request to the AP is inputted to the AP, a conversation response to the conversation request is prepared by the AP, the result of utilization corresponding to the conversation request is prepared by the AP and the result of utilization is inputted through a charging API to a charging system.

(19)日本国特許庁（ＪＰ）　(12)**公開特許公報（Ａ）**　(11)特許出願公開番号
特開2001－312324
（P2001－312324A）
(43)公開日　平成13年11月９日(2001.11.9)

(51)Int.Cl.[7]	識別記号	ＦＩ		テーマコード（参考）
Ｇ０６Ｆ　1/00		Ｇ０６Ｆ　13/00	５３０Ｓ	５Ｂ０４９
13/00	５３０	15/00	３３０Ｂ	５Ｂ０７６
15/00	３３０	17/60	３３２	５Ｂ０８５
17/60	３３２	9/06	６６０Ａ	

審査請求　有　　請求項の数20　ＯＬ　（全 24 頁）

(21)出願番号　特願2000－131796(P2000－131796)

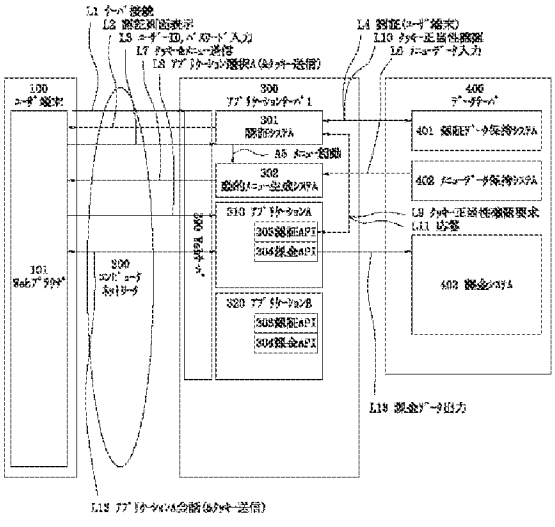
(22)出願日　平成12年４月28日(2000.4.28)

(71)出願人　390001041
エヌイーシーネクサソリューションズ株式会社
東京都港区三田１丁目４番28号
(72)発明者　森田　将夫
東京都港区三田１丁目４番28号　日本電気情報サービス株式会社内
(74)代理人　100095740
弁理士　開口　宗昭

Ｆターム(参考)　5B049 AA02
5B076 FA20 FB05
5B085 AA08 AC04 AE02 AE23

(54)【発明の名称】　ユーザの認証及び課金方法、その記録媒体、アプリケーション・サービスを提供する方法及びシステム

(57)【要約】
【課題】　ベンダーがアプリケーション・サービス・プロバイダの事業に容易に参入できるアプリケーション・サービスを提供する方法及びシステムを提供することを課題とする。
【解決手段】　予め認証された認証証明書が、アプリケーションプログラムに入力され、予め認証された認証証明書が、認証ＡＰＩを介して認証システムに入力され、再度認証された認証証明書が、認証ＡＰＩを介してアプリケーションプログラムに出力され、アプリケーションプログラムに対する会話要求が、アプリケーションプログラムに入力され、会話要求に対する会話応答が、アプリケーションプログラムによって作成され、会話要求に対する利用実績が、アプリケーションプログラムによって作され、利用実績が、課金ＡＰＩを介して課金システムに入力されることによって上記課題を解決する。



1

【特許請求の範囲】

【請求項１】　認証システムによって予め認証された認証証明書が、アプリケーションプログラムに入力され、前記予め認証された認証証明書が、認証ＡＰＩ(Application Program Interface)を介して前記認証システムに入力され、且つ前記認証システムによって再度認証された認証証明書が、前記認証ＡＰＩを介して前記アプリケーションプログラムに出力されるステップを含むこと特徴とする認証関数プログラムによるユーザの認証方法。

【請求項２】　請求項１に記載のステップを実行させるための認証関数プログラムを記録したコンピュータ読み取り可能な記録媒体。

【請求項３】　アプリケーションプログラムに対する会話要求が、前記アプリケーションプログラムに入力される第１のステップと、前記会話要求に対する会話応答が、前記アプリケーションプログラムによって作成される第２のステップと、前記会話要求に対する利用実績が、前記アプリケーションプログラムによって作成される第３のステップと、前記利用実績が、課金ＡＰＩを介して課金システムに入力される第４のステップと、を含むこと特徴とする課金関数プログラムによるユーザの課金方法。

【請求項４】　前記第１のステップ後前記第２のステップ前に処理開始時刻が、前記アプリケーションプログラムによって作成されるステップと、前記第２のステップ後前記第３のステップ前に処理終了時刻が、前記アプリケーションプログラムによって作成されるステップと、を含み、前記第３のステップは、前記処理開始時刻及び前記処理終了時刻に基づいて前記利用実績が作成されることを特徴とする請求項３に記載の課金関数プログラムによるユーザの課金方法。

【請求項５】　前記第３のステップは、前記会話要求及び前記会話応答に基づく課金項目に基づき前記利用実績が作成されることを特徴とする請求項３又は請求項４に記載の課金関数プログラムによるユーザの課金方法。

【請求項６】　請求項３から請求項５の何れか一に記載の全てのステップを実行させるための課金関数プログラムを記録したコンピュータ読み取り可能な記録媒体。

【請求項７】　認証システムによって予め認証された認証証明書が、ユーザ端末からアプリケーションプログラムに受信され、前記予め認証された認証証明書が、前記認証ＡＰＩを介して前記認証システムに入力され、且つ前記認証システムによって再度認証された認証証明書が、前記認証ＡＰＩを介して前記アプリケーションプログラムからユーザ端末に送信されるステップとを含むこと特徴とするアプリケーションプログラムによるユーザの認証方法。

【請求項８】　アプリケーションプログラムに対する会話要求が、ユーザ端末から前記アプリケーションプログラムに受信される第１のステップと、前記会話要求に対する会話応答が、前記アプリケーションプログラムによって作成される第２のステップと、前記会話要求に対する利用実績が、前記アプリケーションプログラムによって作成される第３のステップと、前記利用実績が、課金ＡＰＩを介して前記アプリケーションプログラムから課金システムに入力される第４のステップと、を含むこと特徴とするアプリケーションプログラムによるユーザの課金方法。

【請求項９】　前記第１のステップ後前記第２のステップ前に処理開始時刻が、前記アプリケーションプログラムによって作成されるステップと、前記第２のステップ後前記第３のステップ前に処理終了時刻が、前記アプリケーションプログラムによって作成されるステップと、を含み、前記第３のステップは、前記処理開始時刻及び前記処理終了時刻に基づいて前記利用実績が作成されることを特徴とする請求項８に記載のアプリケーションプログラムによるユーザの課金方法。

【請求項１０】　前記第３のステップは、前記会話要求及び前記会話応答に基づく課金項目に基づき前記利用実績が作成されることを特徴とする請求項８又は請求項９に記載のアプリケーションプログラムによるユーザの課金方法。

【請求項１１】　請求項７に記載のステップと、請求項８から請求項１０の何れか一に記載の全てのステップと、を含むことを特徴とするユーザの課金及び課金方法。

【請求項１２】　請求項７から請求項１１の何れか一に記載の全てのステップを実行させるためのアプリケーションプログラムを記録したコンピュータ読み取り可能な記録媒体。

【請求項１３】　ユーザ識別が、ユーザ端末からアプリケーションサーバに格納される認証システムに受信されるログインステップと、前記ユーザ識別が、前記認証システムによって認証され、前記認証システムによって予め認証された認証証明書が、前記認証システムから前記ユーザ端末に送信される認証ステップと、アプリケーションプログラムに対する会話要求と前記予め認証された認証証明書とが、前記ユーザ端末から前記アプリケーションプログラムに受信される第１のステップと、前記予め認証された認証証明書が、前記アプリケーションプログラムに組み込まれた認証ＡＰＩを介して前記認証システムに入力され、前記予め認証された認証証明書が、前記認証システムによって再度認証され、前記認証システムによって再度認証された認証証明書が、認証ＡＰＩを介して前記アプリケーションプログラムに出力される再認証ステップと、前記会話要求に対する会話応答が、前記アプリケーションプログラムによって作成される第２のステップと、前記会話要求に対する利用実績が、前記アプリケーションプログラムによって作成される第３のステップと、前記再度認証された認証証明書と前記会話

応答とが、前記アプリケーションプログラムからユーザ端末に送信されるステップと、前記利用実績が、課金ＡＰＩを介して前記アプリケーションプログラムからサーバに格納される課金システムに入力される第４のステップと、を含むこと特徴とするアプリケーションサーバによるユーザの認証及び課金方法。

【請求項１４】　請求項１３に記載の全てのステップを実行させるためのアプリケーションサーバに格納されるアプリケーションプログラム及び認証システムプログラムを記録したコンピュータ読み取り可能な記録媒体。

【請求項１５】　ユーザ識別が、ユーザ端末からアプリケーションサーバに格納される認証システムに送受信されるログインステップと、前記ユーザ識別が、前記認証システムによって認証され、前記認証システムによって予め認証された認証証明書が、前記認証システムから前記ユーザ端末に送信される認証ステップと、アプリケーションプログラムに対する会話要求と前記予め認証された認証証明書とが、前記ユーザ端末から前記アプリケーションプログラムに送受信される第１のステップと、前記予め認証された認証証明書が、前記アプリケーションプログラムに組み込まれた認証ＡＰＩを介して前記認証システムに入力され、前記予め認証された認証証明書が、前記認証システムによって再度認証され、前記認証システムによって再度認証された認証証明書が、認証ＡＰＩを介して前記アプリケーションプログラムに出力される再認証ステップと、前記会話要求に対する会話応答が、前記アプリケーションプログラムによって作成される第２のステップと、前記会話要求に対する利用実績が、前記アプリケーションプログラムによって作成される第３のステップと、前記再度認証された認証証明書と前記会話応答とが、前記アプリケーションプログラムからユーザ端末に送受信されるステップと、前記利用実績が、課金ＡＰＩを介して前記アプリケーションプログラムからサーバに格納される課金システムに入力される第４のステップと、を含むこと特徴とするアプリケーション・サービスを提供する方法。

【請求項１６】　前記再認証ステップは、前記予め認証された認証証明書と認証保持システムに予め格納された認証証明書とが前記認証システムによって比較されて一致する場合に、前記予め格納される認証証明書が、新たな認証証明書に前記認証システムによって変更され、且つ前記新たな認証証明書が、前記サーバに格納される認証保持システムに格納されるステップを含み、前記再度認証された認証証明書が、前記新たな認証証明書として出力されることを特徴とする請求項１５に記載のアプリケーション・サービスを提供する方法。

【請求項１７】　前記認証ステップに係る認証証明書が、前記認証システムによって予め暗号化された暗号化済認証証明書であり、前記再認証ステップは、前記予め認証された認証証明書と認証保持システムに予め格納された認証証明書とが前記認証システムによって比較されて一致する場合に、前記予め格納される認証証明書が、新たな暗号化済認証証明書に前記認証システムによって暗号化され、且つ前記新たな暗号化済認証証明書が、前記サーバに格納される認証保持システムに格納されるステップを含み、前記再度認証された認証証明書が、前記新たな暗号化済認証証明書として出力されることを特徴とする請求項１５又は請求項１６に記載のアプリケーション・サービスを提供する方法。

【請求項１８】　ユーザ識別をユーザ端末から受信し、前記ユーザ識別を認証して予め認証された認証証明書をユーザ端末に送信し、前記予め認証された認証証明書をアプリケーションプログラムから前記アプリケーションプログラムに組み込まれた認証ＡＰＩを介して入力され、前記予め認証された認証証明書を再度認証して再度認証された認証証明書を前記アプリケーションプログラムに前記認証ＡＰＩを介して出力する認証システムと、前記会話要求と前記予め認証された認証証明書とを前記ユーザ端末から受信し、前記予め認証された認証証明書を前記認証システムに入力し、前記再度認証された認証証明書を前記認証システムから出力され、前記会話要求に対する会話応答と利用実績とを作成し、前記前記再度認証された認証証明書と前記会話返答とを前記ユーザ端末に送信し、前記利用実績を課金ＡＰＩを介してサーバに格納される課金システムに入力するアプリケーションプログラムと、備えることを特徴とするアプリケーションサーバ。

【請求項１９】　アプリケーションサーバに格納される認証システムによって予め認証された認証証明書を前記認証システムから格納され、前記認証システムによって再度認証された認証証明書を前記認証システムから格納される認証保持システムと、利用実績を前記アプリケーションサーバに格納されるアプリケーションから前記アプリケーションプログラムに組み込まれる認証ＡＰＩを介して入力される課金システムと、を備えるサーバ。

【請求項２０】　請求項１８に記載のアプリケーションサーバと、請求項１９に記載のサーバと、から構成されてなるアプリケーション・サービスを提供するシステム。

【発明の詳細な説明】

【０００１】

【発明の属する技術分野】本発明は、ユーザの認証及び課金方法、その記録媒体、アプリケーション・サービスを提供する方法及びシステムに関するものである。

【０００２】

【発明の背景】アプリケーション・サービスを提供するとは、コンピュータネットワークを介してアプリケーションプログラムを有償でユーザに利用させること、若しくはコンピュータネットワークを介していわゆる商用サービスをユーザに提供することをいう。

5

【０００３】ここで、コンピュータネットワークとは、例えば、インターネット(Internet)、広域通信網（ＷＡＮ:Wide Area Network）等である。インターネットとは、良く知られるように、通信プロトコルＴＣＰ／ＩＰ（Transmission Control Protocol/Internet Protocol）を用いて全世界のネットワークを相互に接続した巨大なコンピュータネットワークである。また、ＴＣＰ／ＩＰとはインターネット、イントラネットで標準的に使われるプロトコルであり、ＴＣＰ／ＩＰはＨＴＴＰ（HyperText Transfer Protocol）、ＦＴＰ（File Transfer Protocol）等の基盤となるプロトコルである。尚、ＨＴＴＰとはＷｅｂサーバとＷｅｂブラウザとの間で（ＨＴＭＬ文書、画像ファイル、音楽ファイル、映像ファイル等）を送受信するのに使われるプロトコルであり、ＨＴＴＰは１つのＨＴＭＬ文書毎にＴＣＰ／ＩＰの接続・転送（ＨＴＭＬ文書のほか、このＨＴＭＬ文書に関連付けられている画像ファイル等）・切断を行うステートレスのプロトコルである。ＨＴＴＰは、IETF(Internet Engineering Task Force)が公開している技術文書rfc(Request For Comments)2616で定義されている(http://www.ietf.org/rfc/rfc2616.txt)。

【０００４】また、ＷｅｂサーバとはＷｅｂブラウザからのリクエストに応じてサーバ上のリソース（ＨＴＭＬ文書、画像ファイル、音楽ファイル、映像ファイル等）をＷｅｂブラウザに送信するものである。ここで、Ｗｅｂサーバには、サーバ上のＨＴＭＬ文書等を単に送信するもののほか、サーバ上のアプリケーションプログラムを実行しその出力結果に基づいて生成されたＨＴＭＬ文書等を送信するものも含む。尚、アプリケーションプログラムの出力結果に基づいてＨＴＭＬ文書を生成するアプリケーションプログラムは、例えば、Ｐｅｒｌ(Practical Extraction and Report Language)、Ｊａｖａ（登録商標）等のプログラム言語で作成されている。Ｐｅｒｌで作成されたアプリケーションプログラムは、良く知られるように、ＣＧＩ（Common Gateway Interface）技術を用いて動的なＨＴＭＬ文書を生成できる。一方、Ｗｅｂブラウザとはサーバ上のリソースをＷｅｂブラウザから受信してリソースを解析してユーザ端末のディスプレイ又はスピーカに表示又は再生するものである。加えて、Ｗｅｂブラウザはユーザからの入力を受け取ってＷｅｂサーバに送信することもできる。ここで、Ｗｅｂブラウザには、Netscape Communications社のNetscape Navigator(Communicator)、Microsoft社のInternet Explorer等がある。

【０００５】ところで、現在、アプリケーション・サービスを提供する事業者、即ちアプリケーション・サービス・プロバイダ（ＡＳＰ:Application Service Provider）と言えば、自らアプリケーションプログラムを開発するベンダーである。即ち、ベンダー自らがアプリケーションプログラムを開発するとともに、業務プロセスプログラム（ユーザ認証プログラム、課金処理プログラム等）を開発して、業務プロセスプログラムを組み込んだアプリケーションプログラムをコンピュータネットワークを介して有償でユーザに利用させている。

【発明が解決しようとする課題】現在のアプリケーション・サービスを提供する方法及びシステムには次のような問題があった。

【０００６】ベンダー自ら業務プロセスプログラム（ユーザ認証プログラム、課金処理プログラム等）を開発する必要があるので、業務プロセスプログラムの開発に必要な設備（プログラム開発者、プログラム開発用サーバ等）を導入する必要がある。即ち、ベンダー自らが業務プロセスプログラムの開発に必要な設備を導入し、業務プロセスを運用する必要がある。この設備の導入コスト及び業務プロセスの運用コストは高額であるので、事実上多くの資本を有する大手ベンダーのみがアプリケーション・サービス・プロバイダ（ＡＳＰ）となっている。従って、ベンダーがアプリケーション・サービス・プロバイダの事業に容易に参入できないという問題があった。

【０００７】また、業務プロセスプログラムの開発に必要な設備の導入コスト及び業務プロセスの運用コストは高額であるので、大手ベンダーでさえアプリケーションプログラムを高価格でユーザに利用させて、初めて導入コスト及び運用コストの採算が取れているのが現状である。従って、ユーザが利用可能なアプリケーションプログラムの数若しくは種類が事実上少ないという問題があった。

【０００８】更に、大手ベンダーが個々にアプリケーション・サービス・プロバイダ（ＡＳＰ）となるので、ユーザはアプリケーション・サービス・プロバイダ（ＡＳＰ）毎に予め契約を結び、且つアプリケーションプログラムの利用代金を支払う必要がある。従って、ユーザは契約、利用代金の支払い等の事務手続が煩雑であるという問題があった。

【０００９】本発明は現在のアプリケーション・サービスを提供する方法及びシステムにおける問題に鑑みてなされたものであって、ベンダーがアプリケーション・サービス・プロバイダの事業に容易に参入できるユーザの認証及び課金方法、その記録媒体、アプリケーション・サービスを提供する方法及びシステムを提供することを課題とする。

【課題を解決するための手段】前記課題を解決する請求項１は、認証システムによって予め認証された認証証明書が、アプリケーションプログラムに入力され、前記予め認証された認証証明書が、認証ＡＰＩ(Application Program Interface)を介して前記認証システムに入力され、且つ前記認証システムによって再度認証された認証証明書が、前記認証ＡＰＩを介して前記アプリケーションプログラムに出力されるステップを含むこと特徴とす

る認証関数プログラムによるユーザの認証方法である。

【００１０】また請求項２は、請求項１に記載のステップを実行させるための認証関数プログラムを記録したコンピュータ読み取り可能な記録媒体である。

【００１１】したがって請求項１又は請求項２の発明によれば、アプリケーションプログラムに認証ＡＰＩ(Application Program Interface)を組み込むだけで、アプリケーションプログラムはユーザの管理ができる。即ち、アプリケーションプログラムを開発するだけで、ユーザ認証プログラムを開発する必要がなくなる。これにより、ベンダーがアプリケーション・サービス・プロバイダの事業に容易に参入できるという利点がある。

【００１２】また請求項３は、アプリケーションプログラムに対する会話要求が、前記アプリケーションプログラムに入力される第１のステップと、前記会話要求に対する会話応答が、前記アプリケーションプログラムによって作成される第２のステップと、前記会話要求に対する利用実績が、前記アプリケーションプログラムによって作成される第３のステップと、前記利用実績が、課金ＡＰＩを介して課金システムに入力される第４のステップと、を含むこと特徴とする課金関数プログラムによるユーザの課金方法である。

【００１３】したがって請求項３の発明によれば、アプリケーションプログラムに課金ＡＰＩ(Application Program Interface)を組み込むだけで、アプリケーションプログラムはユーザの課金ができる。即ち、アプリケーションプログラムを開発するだけで、課金処理プログラムを開発する必要がなくなる。これにより、ベンダーがアプリケーション・サービス・プロバイダの事業に容易に参入できるという利点がある。

【００１４】また請求項４は、前記第１のステップ後前記第２のステップ前に処理開始時刻が、前記アプリケーションプログラムによって作成されるステップと、前記第２のステップ後前記第３のステップ前に処理終了時刻が、前記アプリケーションプログラムによって作成されるステップと、を含み、前記第３のステップは、前記処理開始時刻及び前記処理終了時刻に基づいて前記利用実績が作成されることを特徴とする請求項３に記載の課金関数プログラムによるユーザの課金方法である。

【００１５】また請求項５は、前記第３のステップは、前記会話要求及び前記会話応答に基づく課金項目に基づき前記利用実績が作成されることを特徴とする請求項３又は請求項４に記載の課金関数プログラムによるユーザの課金方法である。

【００１６】また請求項６は、請求項３から請求項５の何れか一に記載の全てのステップを実行させるための課金関数プログラムを記録したコンピュータ読み取り可能な記録媒体である。

【００１７】したがって請求項４から請求項６の何れか一の発明によれば、アプリケーションプログラムはユーザの利用態様に応じて課金ができる。

【００１８】また請求項７は、認証システムによって予め認証された認証証明書が、ユーザ端末からアプリケーションプログラムに受信され、前記予め認証された認証証明書が、前記認証ＡＰＩを介して前記認証システムに入力され、且つ前記認証システムによって再度認証された認証証明書が、前記認証ＡＰＩを介して前記アプリケーションプログラムからユーザ端末に送信されるステップとを含むこと特徴とするアプリケーションプログラムによるユーザの認証方法である。

【００１９】また請求項８は、アプリケーションプログラムに対する会話要求が、ユーザ端末から前記アプリケーションプログラムに受信される第１のステップと、前記会話要求に対する会話応答が、前記アプリケーションプログラムによって作成される第２のステップと、前記会話要求に対する利用実績が、前記アプリケーションプログラムによって作成される第３のステップと、前記利用実績が、課金ＡＰＩを介して前記アプリケーションプログラムから課金システムに入力される第４のステップと、を含むこと特徴とするアプリケーションプログラムによるユーザの課金方法である。

【００２０】また請求項９は、前記第１のステップ後前記第２のステップ前に処理開始時刻が、前記アプリケーションプログラムによって作成されるステップと、前記第２のステップ後前記第３のステップ前に処理終了時刻が、前記アプリケーションプログラムによって作成されるステップと、を含み、前記第３のステップは、前記処理開始時刻及び前記処理終了時刻に基づいて前記利用実績が作成されることを特徴とする請求項８に記載のアプリケーションプログラムによるユーザの課金方法である。

【００２１】また請求項１０は、前記第３のステップは、前記会話要求及び前記会話応答に基づく課金項目に基づき前記利用実績が作成されることを特徴とする請求項８又は請求項９に記載のアプリケーションプログラムによるユーザの課金方法である。

【００２２】また請求項１１は、請求項７に記載のステップと、請求項８から請求項１０の何れか一に記載の全てのステップと、を含むことを特徴とするユーザの課金及び課金方法である。

【００２３】また請求項１２は、請求項７から請求項１１の何れか一に記載の全てのステップを実行させるためのアプリケーションプログラムを記録したコンピュータ読み取り可能な記録媒体である。

【００２４】したがって請求項７から請求項１２の何れか一の発明によれば、認証ＡＰＩ及び／又は課金ＡＰＩを組み込んだアプリケーションプログラムを作成するだけで、アプリケーションプログラムはユーザの認証及び／又課金はができる。即ち、認証ＡＰＩ及び／又は課金ＡＰＩを組み込んだアプリケーションプログラムを開発

するだけで、ユーザ認証及び／又は課金処理プログラムを開発する必要がなくなる。これにより、ベンダーがアプリケーション・サービス・プロバイダの事業に容易に参入できるという利点がある。

【００２５】また請求項１３は、ユーザ識別が、ユーザ端末からアプリケーションサーバに格納される認証システムに受信されるログインステップと、前記ユーザ識別が、前記認証システムによって認証され、前記認証システムによって予め認証された認証証明書が、前記認証システムから前記ユーザ端末に送信される認証ステップと、アプリケーションプログラムに対する会話要求と前記予め認証された認証証明書とが、前記ユーザ端末から前記アプリケーションプログラムに受信される第１のステップと、前記予め認証された認証証明書が、前記アプリケーションプログラムに組み込まれた認証ＡＰＩを介して前記認証システムに入力され、前記予め認証された認証証明書が、前記認証システムによって再度認証され、前記認証システムによって再度認証された認証証明書が、認証ＡＰＩを介して前記アプリケーションプログラムに出力される再認証ステップと、前記会話要求に対する会話応答が、前記アプリケーションプログラムによって作成される第２のステップと、前記会話要求に対する利用実績が、前記アプリケーションプログラムによって作成される第３のステップと、前記再度認証された認証証明書と前記会話応答とが、前記アプリケーションプログラムからユーザ端末に送信されるステップと、前記利用実績が、課金ＡＰＩを介して前記アプリケーションプログラムからサーバに格納される課金システムに入力される第４のステップと、を含むこと特徴とするアプリケーションサーバによるユーザの認証及び課金方法である。

【００２６】また請求項１４は、請求項１３に記載の全てのステップを実行させるためのアプリケーションサーバに格納されるアプリケーションプログラム及び認証システムプログラムを記録したコンピュータ読み取り可能な記録媒体である。

【００２７】したがって請求項１３又は請求項１４の発明によれば、認証ＡＰＩ及び課金ＡＰＩを組み込んだアプリケーションプログラムと認証システムとをアプリケーションサーバに格納するだけで、ベンターはアプリケーション・サービスを提供することができる。即ち、ベンダーがアプリケーション・サービス・プロバイダの事業に容易に参入できるという利点がある。

【００２８】また請求項１５は、ユーザ識別が、ユーザ端末からアプリケーションサーバに格納される認証システムに送受信されるログインステップと、前記ユーザ識別が、前記認証システムによって認証され、前記認証システムによって予め認証された認証証明書が、前記認証システムから前記ユーザ端末に送信される認証ステップと、アプリケーションプログラムに対する会話要求と前記予め認証された認証証明書とが、前記ユーザ端末から前記アプリケーションプログラムに送受信される第１のステップと、前記予め認証された認証証明書が、前記アプリケーションプログラムに組み込まれた認証ＡＰＩを介して前記認証システムに入力され、前記予め認証された認証証明書が、前記認証システムによって再度認証され、前記認証システムによって再度認証された認証証明書が、認証ＡＰＩを介して前記アプリケーションプログラムに出力される再認証ステップと、前記会話要求に対する会話応答が、前記アプリケーションプログラムによって作成される第２のステップと、前記会話要求に対する利用実績が、前記アプリケーションプログラムによって作成される第３のステップと、前記再度認証された認証証明書と前記会話応答とが、前記アプリケーションプログラムからユーザ端末に送受信されるステップと、前記利用実績が、課金ＡＰＩを介して前記アプリケーションプログラムからサーバに格納される課金システムに入力される第４のステップと、を含むこと特徴とするアプリケーション・サービスを提供する方法である。

【００２９】したがって請求項１５の発明によれば、認証ＡＰＩ及び課金ＡＰＩを組み込んだアプリケーションプログラムと認証システムとをアプリケーションサーバに格納するだけで、ベンターはアプリケーション・サービスを提供することができる。即ち、ベンダーがアプリケーション・サービス・プロバイダの事業に容易に参入できるという利点がある。

【００３０】また、ベンダーはユーザ認証及び課金処理プログラムを開発する必要がなくなるので、新たなアプリケーションプログラムを開発することができる。この場合、この新たなアプリケーションプログラムをアプリケーションサーバに格納するだけで、ベンターは複数のアプリケーション・サービスを提供することができる。即ち、ユーザが利用可能なアプリケーションプログラムの数若しくは種類が多くなるという利点がある。ここで、複数のアプリケーションプログラムがそれぞれ認証ＡＰＩを介して一の認証システムによって、ユーザを識別することができる。換言すれば、複数のアプリケーションプログラム間でユーザ管理の統一性を実現することができる。

【００３１】更に、別のベンダーが別のアプリケーションプログラムを開発した場合、複数のアプリケーションプログラムと一の認証システムとをアプリケーションサーバに別のアプリケーションプログラムを格納するだけで、別のベンダーもアプリケーション・サービスを提供することができる。即ち、アプリケーションサーバは複数のベンダーのアプリケーションプログラムをユーザに有償で利用させることができる。ここで、複数のベンダーのアプリケーションプログラムがそれぞれ課金ＡＰＩを介して一の課金システムによって、利用実績を集計して利用金額を算出することができる。換言すれば、複数

のベンダーの間で課金管理の統一性を実現することができる。課金システムはユーザ別に利用金額を算出してユーザに支払いを求める一方、課金システムはベンダー別に利用金額を算出してベンダーに支払う。これにより、ユーザは利用代金の支払い等の事務手続が煩雑でないという利点がある。

【００３２】また請求項１６は、前記再認証ステップは、前記予め認証された認証証明書と認証保持システムに予め格納された認証証明書とが前記認証システムによって比較されて一致する場合に、前記予め格納される認証証明書が、新たな認証証明書に前記認証システムによって変更され、且つ前記新たな認証証明書が、前記サーバに格納される認証保持システムに格納されるステップを含み、前記再度認証された認証証明書が、前記新たな認証証明書として出力されることを特徴とする請求項１５に記載のアプリケーション・サービスを提供する方法である。

【００３３】また請求項１７は、前記認証ステップに係る認証証明書が、前記認証システムによって予め暗号化された暗号化済認証証明書であり、前記再認証ステップは、前記予め認証された認証証明書と認証保持システムに予め格納された認証証明書とが前記認証システムによって比較されて一致する場合に、前記予め格納される認証証明書が、新たな暗号化済認証証明書に前記認証システムによって暗号化され、且つ前記新たな暗号化済認証証明書が、前記サーバに格納される認証保持システムに格納されるステップを含み、前記再度認証された認証証明書が、前記新たな暗号化済認証証明書として出力されることを特徴とする請求項１５又は請求項１６に記載のアプリケーション・サービスを提供する方法である。

【００３４】したがって請求項１６又は請求項１７の発明によれば、認証証明書の偽造を防止することができるという利点がある。

【００３５】また請求項１８は、ユーザ識別をユーザ端末から受信し、前記ユーザ識別を認証して予め認証された認証証明書をユーザ端末に送信し、前記予め認証された認証証明書をアプリケーションプログラムから前記アプリケーションプログラムに組み込まれた認証ＡＰＩを介して入力され、前記予め認証された認証証明書を再度認証して再度認証された認証証明書を前記アプリケーションプログラムに前記認証ＡＰＩを介して出力する認証システムと、前記会話要求と前記予め認証された認証証明書とを前記ユーザ端末から受信し、前記予め認証された認証証明書を前記認証システムに入力し、前記再度認証された認証証明書を前記認証システムから出力され、前記会話要求に対する会話応答と利用実績とを作成し、前記前記再度認証された認証証明書と前記会話返答とを前記ユーザ端末に送信し、前記利用実績を課金ＡＰＩを介してサーバに格納される課金システムに入力するアプリケーションプログラムと、備えることを特徴とするアプリケーションサーバである。

【００３６】また請求項１９は、アプリケーションサーバに格納される認証システムによって予め認証された認証証明書を前記認証システムから格納され、前記認証システムによって再度認証された認証証明書を前記認証システムから格納される認証保持システムと、利用実績を前記アプリケーションサーバに格納されるアプリケーションから前記アプリケーションプログラムに組み込まれる認証ＡＰＩを介して入力される課金システムと、を備えるサーバである。

【００３７】また請求項２０は、請求項１８に記載のアプリケーションサーバと、請求項１９に記載のサーバと、から構成されてなるアプリケーション・サービスを提供するシステムである。

【００３８】したがって請求項１８から請求項２０何れか一の発明によれば、認証ＡＰＩ及び課金ＡＰＩを組み込んだアプリケーションプログラムと認証システムとをアプリケーションサーバに格納するだけで、ベンターはアプリケーション・サービスを提供することができる。即ち、ベンダーがアプリケーション・サービス・プロバイダの事業に容易に参入できるという利点がある。

【００３９】また、ベンダーはユーザ認証及び課金処理プログラムを開発する必要がなくなるので、新たなアプリケーションプログラムを開発することができる。この場合、この新たなアプリケーションプログラムをアプリケーションサーバに格納するだけで、ベンターは複数のアプリケーション・サービスを提供することができる。即ち、ユーザが利用可能なアプリケーションプログラムの数若しくは種類が多くなるという利点がある。ここで、複数のアプリケーションプログラムがそれぞれ認証ＡＰＩを介して一の認証システムによって、ユーザを識別することができる。換言すれば、複数のアプリケーションプログラム間でユーザ管理の統一性を実現することができる。

【００４０】更に、別のベンダーが別のアプリケーションプログラムを開発した場合、複数のアプリケーションプログラムと一の認証システムとをアプリケーションサーバに別のアプリケーションプログラムを格納するだけで、別のベンダーもアプリケーション・サービスを提供することができる。即ち、アプリケーションサーバは複数のベンダーのアプリケーションプログラムをユーザに有償で利用させることができる。ここで、複数のベンダーのアプリケーションプログラムがそれぞれ課金ＡＰＩを介して一の課金システムによって、利用実績を集計して利用金額を算出することができる。換言すれば、複数のベンダーの間で課金管理の統一性を実現することができる。課金システムはユーザ別に利用金額を算出してユーザに支払いを求める一方、課金システムはベンダー別に利用金額を算出してベンダーに支払う。これにより、ユーザは利用代金の支払い等の事務手続が煩雑でないと

いう利点がある。

【００４１】尚、現在のアプリケーション・サービスを提供する方法及びシステムには次のような問題もあった。

【００４２】例えば、複数のユーザが同一のアプリケーションプログラムを同時に利用することも当然に生じ得る。この場合、アプリケーションプログラムは複数のユーザのそれぞれに対するＨＴＭＬ文書等のを生成する必要があるので、アプリケーションプログラムの処理速度が低下することになる。

【００４３】ここで、アプリケーションプログラムの処理速度の低下を防ぐためには、アプリケーションプログラムが格納されるサーバ（以下、「アプリケーションサーバ」とも言う。）の処理能力を高くすれば解決する。ところが、アプリケーションサーバのコストは処理能力が高くなるに伴って飛躍的に高くなるのが現状である。具体的に言えば、アプリケーションサーバのコストは、演算速度に依存するＣＰＵ(Central Processing Unit)のクロック周波数、の送信速度に依存するルータの通信速度等が２倍になると２倍以上になることは言うまでもない。また、現在の技術ではＣＰＵのクロック周波数等には上限があるのが現状である。従って、アプリケーションサーバの処理能力を高くすることには限界があるので、ある程度の数のユーザが同一のアプリケーションプログラムを同時に利用する場合には、アプリケーションプログラムの処理速度の低下を防ぐことができるが、非常に多くの数のユーザが同一のアプリケーションプログラムを同時に利用する場合、アプリケーションプログラムの処理速度が低下することになる。

【００４４】また、アプリケーションプログラムの処理速度の低下を防ぐためには、アプリケーションサーバを複数台設置することによっても解決する。即ち、アプリケーションプログラムを複数のアプリケーションサーバに分散させて、一台のアプリケーションサーバに格納されるアプリケーションプログラムを利用するユーザの数を分散させる。これにより、アプリケーションプログラムの処理速度の低下を防止することができる。結果として、例えば、同一のユーザが同一のアプリケーションプログラムを複数のアプリケーションサーバで利用することも当然に生じ得る。即ち、同一のユーザが同一のアプリケーションプログラムをあるアプリケーションサーバで利用し始め、利用を一旦中止する（一時退避）。しかる後に同一のユーザが同一のアプリケーションプログラムを別のアプリケーションサーバで利用を再開することも当然に生じ得る。この場合、業務プロセスプログラムのそれぞれがアプリケーションプログラム（ユーザ認証プログラム、課金処理プログラム等）に組み込まれているので、アプリケーションサーバ単位で同一のユーザのユーザ認証がされ、同一のユーザに課金処理がされることになる。

【００４５】従って、アプリケーション・サービス・プロバイダ（ＡＳＰ）が同一のユーザに対して利用代金の支払いを求める場合、アプリケーション・サービス・プロバイダを行う事業者は同一のユーザに対する課金処理の結果をアプリケーションサーバのそれぞれで収集して利用代金の額を算出することになる。尚、課金処理の結果である利用代金の額は、アプリケーション・サービス・プロバイダを行う事業者とユーザとの契約でユーザが利用したアプリケーションプログラムの実績（利用時間、利用回数等）に基づいて算出されている。

【００４６】ここで、アプリケーション・サービス・プロバイダ（ＡＳＰ）とユーザとの間の契約で利用代金の額が、ユーザの利用時間（実績）に線形である場合、アプリケーションサーバ単位の課金処理の結果である利用代金の額を単に足すだけでよい。一例を挙げれば、契約で利用代金の額が、１０００円／１時間当たり（ユーザの利用時間）であるとする。Ａ（ユーザ）さんがＡ（アプリケーションプログラム）をＡサーバ（アプリケーションサーバ）で１時間利用し、しかる後にＡさんがＡをＢサーバで２時間利用したとする。Ａサーバは課金処理の結果として利用代金の額を１０００円であると算出し、Ｂサーバは課金処理の結果として利用代金の額を２０００円であると算出する。これに対し、課金処理の結果である利用代金の額のそれぞれを単に足すだけでよい（１０００円＋２０００円）。

【００４７】しかしながら、アプリケーション・サービス・プロバイダ（ＡＳＰ）とユーザとの間の契約で利用代金の額が、ユーザの利用時間（実績）に線形でない場合、アプリケーションサーバ単位の課金処理の結果である利用代金の額を単に足すことはできない。一例を挙げれば、契約で利用代金の額が、合計利用時間が１時間までは１０００円／１時間当たりであり、合計利用時間が１時間以降２時間までは２０００円／１時間当たりであるとする。先程と同様に、ＡさんがＡをＡサーバで１時間利用し、しかる後にＡさんがＡをＢサーバで２時間利用したとする。Ａサーバは課金処理の結果として利用代金の額を１０００円であると算出し、Ｂサーバは課金処理の結果として利用代金の額を３０００円（１０００円＋２０００円）であると算出する。これに対し、課金処理の結果である利用代金の額のそれぞれを単に足すと４０００円（１０００円＋３０００円）。一方、契約ではＡさんがＡを３時間利用すると、契約の利用代金の額は５０００円（１０００×１＋２０００×２）である。

【００４８】即ち、契約の利用代金の額がユーザの利用時間（実績）に線形でない場合、アプリケーションサーバ単位の課金処理の結果である利用代金の額を単に足すことはできない。従って、ＡさんはＡをＡサーバで１時間利用し、しかる後にＡさんはＡを再びＡサーバで２時間利用しなければならない。即ち、同一のユーザが同一のアプリケーションプログラムを同一のアプリケーショ

ンサーバで利用しなければならない。結果として、アプリケーションプログラムを複数のアプリケーションサーバに分散させても、一台のアプリケーションサーバに格納されるアプリケーションプログラムを利用するユーザの数を分散できない場合がある。これにより、アプリケーションプログラムの処理速度が低下することになる。

【００４９】また請求項２１の発明は、複数のアプリケーションサーバに格納されるアプリケーションプログラムのそれぞれのうち一のアプリケーションプログラムに対する会話要求と予め認証された認証証明書とが、ユーザ端末から前記一のアプリケーションプログラムが格納される一のアプリケーションサーバに送受信される第１のステップと、前記予め認証された認証証明書とサーバに予め格納される認証証明書とが前記一のアプリケーションサーバによって比較されて一致する場合に、前記会話要求に対する会話返答と利用実績とが、前記一のアプリケーションサーバによって作成され、且つ前記会話返答と前記一のアプリケーションサーバによって再度認証された認証証明書とが、前記一のアプリケーションサーバから前記ユーザ端末に送受信される第２のステップと、前記利用実績が、前記一のアプリケーションサーバから前記サーバに送受信される第３のステップと、を含むことを特徴とするアプリケーション・サービスを提供する方法である。

【００５０】したがって請求項２１の発明によれば、複数のアプリケーションサーバのそれぞれが、受信した認証証明書とサーバに予め格納される認証証明書とを比較して、受信した認証証明書の正当性の確認を行うことができる。即ち、複数のアプリケーションサーバのそれぞれが、同一のユーザを識別することができる。換言すれば、複数のアプリケーションサーバ間でユーザ管理の統一性を実現することができる。これにより、アプリケーションプログラムを複数のアプリケーションサーバに分散させて、アプリケーションプログラムの処理速度の低下を防止することができるという利点がある。

【００５１】また、請求項２２の発明は、前記第３のステップ後に前記複数のアプリケーションサーバに格納されるアプリケーションプログラムのそれぞれのうち他のアプリケーションプログラムに対する会話要求と前記再度認証された認証証明書とが、前記ユーザ端末から前記他のアプリケーションプログラムが格納される他のアプリケーションサーバに送受信される第４のステップと、前記再度認証された認証証明書と前記予め格納された認証証明書とが前記他のアプリケーションサーバによって比較されて一致する場合に、前記会話要求に対する会話返答と利用実績とが、前記他のアプリケーションサーバによって作成され、且つ前記会話返答と前記他のアプリケーションサーバによって再度認証された認証証明書とが、前記他のアプリケーションサーバから前記ユーザ端末に送受信される第５のステップと、前記利用実績と

が、前記他のアプリケーションサーバから前記サーバに送受信される第６のステップと、を含むことを特徴とする第１項に記載のアプリケーション・サービスを提供する方法である。

【００５２】したがって請求項２２の発明によれば、第１項の発明の利点があるとともに、ユーザは一のアプリケーションプログラムによって再度認証された認証証明書を他のアプリケーションプログラムに送信することにより、他のアプリケーションプログラムを利用することができる。結果として、ユーザは同一のアプリケーションプログラムを他のアプリケーションサーバで利用することもでき、また、異なるアプリケーションプログラムを契約の範囲内で自由に利用することができる。このように、ユーザが同一のアプリケーションサーバだけに制限されることがないので、ユーザが複数のアプリケーションサーバに分散することになる。これにより、アプリケーションプログラムの処理速度の低下を防止することができるという利点がある。

【００５３】また、請求項２３の発明は、前記第３のステップ又は前記第６のステップ後に前記一のアプリケーションプログラムに対する会話要求と前記再度認証された認証証明書とが、ユーザ端末から前記一のアプリケーションサーバに送受信される第７のステップと、前記第７のステップ後に前記第２のステップ及び前記第３のステップが、１以上繰り返して実行される第８のステップと、を含むことを特徴とする第１項又は請求項２２に記載のアプリケーション・サービスを提供する方法である。

【００５４】したがって請求項２３の発明によれば、第１項又は請求項２２の発明の利点があるとともに、ユーザは一のアプリケーションサーバから他のアプリケーションサーバに接続した後、再び一のアプリケーションサーバに格納される同一のアプリケーションプログラムを利用することができる。例えば、ユーザが他のアプリケーションサーバに接続を切り換えたとしても、必ずしも他のアプリケーションサーバに格納される他のアプリケーションプログラムの処理速度が速いとは限らない。従って、他のアプリケーションプログラムの処理速度が遅い場合には、ユーザが一のアプリケーションプログラムの利用を再開することにより、他のアプリケーションプログラムの処理速度の更なる低下を防止することができる。

【００５５】また、請求項２４の発明は、前記一のアプリケーションサーバ及び／又は前記他のアプリケーションサーバから受信した利用実績に基づく利用代金の額が、前記サーバによって作成されるステップを含むことを特徴とする第１項から請求項２３の何れか一に記載のアプリケーション・サービスを提供する方法である。

【００５６】したがって請求項２４の発明によれば、第１項から請求項２３の何れか一の発明の利点があると

もに、利用実績がアプリケーションサーバのそれぞれからサーバに送信されるので、契約の利用代金の額がユーザの利用時間（実績）に線形でない場合であっても、サーバで利用実績を収集して、課金処理を行って利用代金の額を算出することができる。即ち、複数のアプリケーションサーバ間で課金管理の統一性を実現することができるという利点がある。

【００５７】また、請求項２５の発明は、アプリケーションプログラムの応答時間が、前記複数のアプリケーションサーバのそれぞれから前記サーバに定期的に送受信される監視ステップと、前記第１のステップ前にアプリケーションプログラムに対する接続要求とユーザ識別とが、前記ユーザ端末から前記複数のアプリケーションサーバのうち何れか一のアプリケーションサーバに送受信されるログインステップと、前記ユーザ識別と前記サーバに予め格納されるユーザ識別とが前記何れか一のアプリケーションサーバによって比較されて一致する場合に、前記第１のステップに係る予め認証された認証証明書が、前記サーバに格納され、前記応答時間のそれぞれが前記サーバから前記何れか一のアプリケーションサーバに送受信されて、前記応答時間のそれぞれうち最も応答時間が短いアプリケーションプログラムが格納される最速アプリケーションサーバを特定し得るメニューが、作成され、且つ前記予め認証された認証証明書と前記メニューとが、前記何れか一のアプリケーションサーバから前記ユーザ端末に送受信される認証ステップと、を含み、前記メニューに基づいて前記最速アプリケーションサーバが前記ユーザ端末から接続されるとともに、前記第１のステップに係る一のアプリケーションサーバが前記最速アプリケーションサーバとして前記第１のステップが実行されることを特徴とする第１項から請求項２４の何れか一に記載のアプリケーション・サービスを提供する方法である。

【００５８】したがって請求項２５の発明によれば、第１項から請求項２４の何れか一の発明の利点があるとともに、アプリケーションプログラムの応答時間が監視されているので、アプリケーションプログラムの処理速度の低下を確実に防止することができるという利点がある。

【００５９】また、請求項２６の発明は、前記第２のステップ及び／又は前記第５のステップは、前記認証証明書と前記予め格納された認証証明書とが比較されて一致する場合に、前記予め格納される認証証明書が、新たな認証証明書に前記一のアプリケーションサーバ及び／又は前記他のアプリケーションサーバによって変更され、且つ前記新たな認証証明書が、前記サーバに格納されるステップを含み、前記再度認証された認証証明書が、前記新たな認証証明書として送受信されることを特徴とする第１項から請求項２５の何れか一に記載のアプリケーション・サービスを提供する方法である。

【００６０】また、請求項２７の発明は、前記第１のステップに係る認証証明書が、予め暗号化された暗号化済認証証明書であり、前記第２のステップ及び／又は前記第５のステップは、前記認証証明書と前記予め格納された認証証明書とが比較されて一致する場合に、前記予め格納された認証証明書が、新たな暗号化済認証証明書に前記一のアプリケーションサーバ及び／又は前記他のアプリケーションサーバによって暗号化され、且つ前記新たな暗号化済認証証明書が、前記サーバに格納されるステップを含み、前記再度認証された認証証明書が、前記新たな暗号化済認証証明書として送受信されることを特徴とする第１項から請求項２６の何れか一に記載のアプリケーション・サービスを提供する方法である。

【００６１】したがって請求項２５又は請求項２６の発明によれば、認証証明書の偽造を防止することができるという利点がある。

【００６２】また、請求項２８の発明は、複数のアプリケーションサーバに格納されるアプリケーションプログラムのそれぞれのうち一のアプリケーションプログラムが格納される一のアプリケーションサーバを選択し、前記一のアプリケーションプログラムに対する会話要求と予め認証された認証証明書とを前記一のアプリケーションサーバに送信し、且つ前記会話要求に対する会話返答と前記一のアプリケーションサーバによって再度認証された認証証明書とを前記一のアプリケーションサーバから受信する手段を備えるユーザ端末と、前記会話要求と前記予め認証された認証証明書とを前記ユーザ端末から受信し、前記予め認証された認証証明書とサーバに予め格納される認証証明書とを比較して一致する場合に、前記会話返答と前記会話要求に対する利用実績とを作成し、前記会話返答と再度認証された認証証明書とを前記ユーザ端末に送信し、且つ利用実績を前記サーバに送信する手段をそれぞれ備える複数のアプリケーションサーバと、前記認証された認証証明書を予め格納し、且つ前記利用実績を前記一のアプリケーションサーバから受信する手段を備えるサーバと、から構成されてなるアプリケーション・サービスを提供するシステムである。

【００６３】したがって請求項２８の発明によれば、複数のアプリケーションサーバのそれぞれが、受信した認証証明書とサーバに予め格納される認証証明書とを比較して、受信した認証証明書の正当性の確認を行うことができる。即ち、複数のアプリケーションサーバのそれぞれが、同一のユーザを識別することができる。換言すれば、複数のアプリケーションサーバ間でユーザ管理の統一性を実現することができる。これにより、アプリケーションプログラムを複数のアプリケーションサーバに分散させて、アプリケーションプログラムの処理速度の低下を防止することができるという利点がある。

【００６４】また、請求項２９の発明は、前記サーバが、前記利用実績に基づく利用代金の額を作成する手段

を備えることを特徴とする請求項２８に記載のアプリケーション・サービスを提供するシステムである。

【００６５】したがって請求項２９の発明によれば、請求項２８の発明の利点があるとともに、利用実績がアプリケーションサーバのそれぞれからサーバに送信されるので、契約の利用代金の額がユーザの利用時間（実績）に線形でない場合であっても、サーバで利用実績を収集して、課金処理を行って利用代金の額を算出することができる。即ち、複数のアプリケーションサーバ間で課金管理の統一性を実現することができるという利点がある。

【００６６】また、請求項３０の発明は、ユーザ端末が、アプリケーションプログラムに対する接続要求とユーザ識別とを前記複数のアプリケーションサーバのうち何れか一のアプリケーションサーバに送信し、且つ前記予め認証された認証証明書と前記応答時間のそれぞれのうち最も応答時間が短いアプリケーションプログラムが格納される最速アプリケーションサーバを特定し得るメニューとを前記何れか一のアプリケーションサーバから受信して、前記一のアプリケーションサーバを前記最速アプリケーションサーバとして選択する手段を備え、前記複数のアプリケーションサーバのそれぞれが、アプリケーションプログラムの応答時間を前記サーバに定期的に送信し、前記接続要求と前記ユーザ識別とを前記ユーザ端末から受信して、前記ユーザ識別とサーバに予め格納されるユーザ識別とを比較して一致する場合に、前記予め認証された認証証明書を前記サーバに格納し、前記応答時間のそれぞれを前記サーバから受信して、前記メニューを作成し、且つ前記予め認証された認証証明書と前記メニューとを前記ユーザ端末に送信する手段を備え、前記サーバが、前記応答時間を前記複数のアプリケーションサーバのそれぞれから定期的に受信し、前記ユーザ識別を予め格納し、且つ前記応答時間のそれぞれを前記アプリケーションサーバのそれぞれに送信する手段を備えることを特徴とする請求項２８又は請求項２９に記載のアプリケーション・サービスを提供するシステムである。

【００６７】したがって請求項３０の発明によれば、請求項２８又は請求項２９の発明の利点があるとともに、アプリケーションプログラムの応答時間が監視されているので、アプリケーションプログラムの処理速度の低下を確実に防止することができるという利点がある。

【００６８】また、請求項３１の発明は、前記複数のアプリケーションサーバのそれぞれが、前記認証証明書と前記予め格納された認証証明書とを比較して一致する場合に、前記予め格納された認証証明書を新たな認証証明書に変更して、前記新たな認証証明書を格納する手段を備え、前記再度認証された認証証明書を前記新たな認証証明書として送信することを特徴とする請求項２８から請求項３０の何れか一に記載のアプリケーション・サービスを提供するシステムである。

【００６９】また、請求項３２の発明は、前記ユーザ端末が送信する前記予め認証された認証証明書が、予め暗号化された暗号化済認証証明書であり、前記複数のアプリケーションサーバのそれぞれが、前記認証証明書と予め格納された認証証明書とを比較して一致する場合に、前記予め格納された認証証明書を新たな暗号化済認証証明書に暗号化して、前記新たな暗号化済認証証明書を格納する手段を備え、前記再度認証された認証証明書を前記新たな暗号化済認証証明書として送信することを特徴とする請求項２８から請求項３１の何れか一に記載のアプリケーション・サービスを提供するシステムである。

【００７０】したがって請求項３１又は請求項３２の発明によれば、認証証明書の偽造を防止することができるという利点がある。

【発明の実施の形態】以下に本発明の実施の形態のユーザの認証及び課金方法、その記録媒体，アプリケーション・サービスを提供する方法及びシステムにつき図面を参照して説明する。

【００７１】（実施の形態１）まず、本発明の実施の形態１のアプリケーション・サービスを提供するシステムにつき、図５を参照して、説明する。図５は本発明の実施の形態１のアプリケーション・サービスを提供するシステムの構成例を示すブロック図である。

【００７２】先ず、図５を参照するに，本実施の形態１のアプリケーション・サービスを提供するシステムは，アプリケーションプログラムを利用するユーザ端末(100)と、アプリケーションプログラムが動作するアプリケーションサーバ１(300)と、各種のを格納するサーバ(500)と、から構成されてなる。ここで、ユーザ端末(100)と、アプリケーションサーバ１(300)とは、コンピュータネットワーク(200)を介して相互に接続可能に構成されている。また、サーバ(500)と、アプリケーションサーバ１(300)とは、主として同一組織内で用いられる総合的な情報通信ネットワーク（ＬＡＮ:Local Area Network）、公衆回線、又は専用線等を介して相互に接続可能に構成されている。尚、ユーザ端末(100)と、アプリケーションサーバ１(300)と、サーバ(500)とが、コンピュータネットワーク(200)を介して相互に接続可能に構成されていてもよい。また、コンピュータネットワーク(200)とは、例えば、インターネット(Internet)、広域通信網（ＷＡＮ:Wide Area Network）等である。更に、コンピュータネットワーク(200)，ＬＡＮ等は、有線、無線を問わず、ユーザ端末(100)とアプリケーションサーバ１(300)との間、ユーザ端末(100)とサーバ(500)との間、アプリケーションサーバ１(300)とサーバ(500)との間、及びユーザ端末(100)とアプリケーションサーバ１(300)とサーバ(500)との間のそれぞれでを送受信するためのものであればよい。

【００７３】ユーザ端末(100)は、コンピュータネット

ワーク(200)を介してホームページへのアクセス等を可能とするＷｅｂブラウザ(101)と、ホームページを表示可能とする表示手段（図示せず）と，文字・記号等を入力するキーボード（ボタン）、マウス等の入力手段（図示せず）と、を備える。ここで、ユーザ端末(100)は、例えば、パソコン（デスクトップ型パソコン、ノート型パソコン）、携帯電話機等である。

【００７４】アプリケーションサーバ１(300)は、ＨＴＭＬ文書等のリソース（図示せず）を格納する記憶手段と、ユーザを認証する認証システム(301)と、ユーザ個別のメニューを動的に生成する動的メニュー生成システム(302)と、コンピュータネットワーク(200)を介して有償で提供する複数のアプリケーションプログラムＡ(310)及びアプリケーションプログラムＢ(320)と、Ｗｅｂサーバ(390)と、を備える。また、アプリケーションプログラムＡ(310)及びアプリケーションプログラムＢ(320)のそれぞれは、認証ＡＰＩ(303)と課金ＡＰＩ(304)と、を備える。更に、Ｗｅｂサーバ(390)は、ユーザ端末(100)のＷｅｂブラウザ(101)とＨＴＴＰで通信する。即ち、Ｗｅｂサーバ(390)は、認証システム(301)、動的メニュー生成システム(302)、アプリケーションプログラムＡ(310)、及びアプリケーションプログラムＢ(320)のそれぞれと、Ｗｅｂブラウザ(101)と、の通信を中継する。ここで、アプリケーションサーバ１(300)は、例えば，いわゆるサーバ（ハード・ディスク等の記憶手段の容量が大きいオフコン）、パソコン等である。

【００７５】サーバ(500)は、ユーザ認証に必要な（ユーザＩＤ、パスワード、端末識別子等を含む。）を格納する認証保持システム(501)と、ユーザごとにあらかじめ合意した利用可能アプリケーションプログラムのリストを格納するメニュー保持システム(502)と、利用実績を蓄積してユーザ別アプリケーションプログラム別請求書を作成する課金システム(503)と、を備える。ここで、サーバ(500)は、例えば、いわゆるサーバ、パソコン等である。

【００７６】尚、本実施の形態１のアプリケーション・サービスを提供するシステムの構成要素のうち、認証システム(301)と、動的メニュー生成システム(302)と、認証ＡＰＩ(303)と、課金ＡＰＩ(304)と、認証保持システム(501)と、メニュー保持システム(502)と、課金システム(503)とをＡＳＰ基盤システムという。また、ＡＳＰ基盤システムを格納するサーバをＡＳＰ基盤サーバ(900)という。

【００７７】次に、本発明の実施の形態のアプリケーション・サービスを提供する方法及びシステムの前提となるアプリケーション・サービスを提供するための準備をするシステム及び方法につき、図１～図４を参照して、説明する。尚、アプリケーション・サービスを提供するための準備をするシステム方法は、アプリケーションプログラムを準備する方法（ステップＡ１～Ａ７）と、アプリケーションプログラムを利用するユーザの登録方法（ステップＢ１～Ｂ５）と、から成る。図１はアプリケーション・サービスを提供するための準備をするシステムの構成例を示すブロック図である。図２はアプリケーションプログラムを準備する方法を説明するためのフローチャート図である。

【００７８】予め、ＡＳＰ基盤システムのうち認証システム(301)と、動的メニュー生成システム(302)と、認証ＡＰＩ(303)と、課金ＡＰＩ(304)とが、アプリケーションサーバ１(300)に格納されている。また、ＡＳＰ基盤システムの残りの認証保持システム(501)と、メニュー保持システム(502)と、課金システム(503)とが、ＡＳＰ基盤サーバ(900)に格納されている。加えて、開発されたアプリケーションプログラムが、ベンダー端末(700,701)に格納されている。

【００７９】[ステップＡ１]

（アプリケーションプログラムを準備する方法）ＡＳＰ基盤サーバ(900)に係るＡＳＰ基盤システム（認証システム(301)、動的メニュー生成システム(302)、認証ＡＰＩ(303)、課金ＡＰＩ(304)）が、公衆回線若しくは専用線(220)、又は光磁気ディスク（ＭＯ:Maqneto Optical disk)、光ディスク(ＣＤ:Compact Disk,ＤＶＤ:Digital Versatile Disk)、フロッピー（登録商標）ディスク(ＦＤ:Floppy Disk)　等の補助記憶媒体(230)を介してＡＳＰ基盤サーバ(900)からベンダー端末(700,710)に提供されて格納される。

【００８０】[ステップＡ２]開発されたアプリケーションプログラムＡをＡＳＰ基盤システムに適合されるため、提供された認証ＡＰＩ(303)及び課金ＡＰＩ(304)が、ベンダー端末(700)によって開発されたアプリケーションプログラムＡ(310)に組み込まれる。

【００８１】[ステップＡ３]アプリケーションサーバ１(300)でアプリケーションプログラムのサービスを行うため、提供された認証システム(301)及び動的メニュー生成システム(302)が、ＬＡＮ(210)介してベンダー端末(700)からアプリケーションサーバ２(400)にインストールされて格納される。尚、ここで、ＡＳＰ基盤システムの認証システム(301)及び動的メニュー生成システム(302)が、公衆回線等(220,230)を介してアプリケーションサーバ２(400)にインストールされてもよいことは言うまでもない。

【００８２】[ステップＡ４]認証ＡＰＩ(303)及び課金ＡＰＩ(304)が組み込まれたアプリケーションプログラムＡ(310)が、ベンダー端末(700)からアプリケーションサーバ１(300)にＬＡＮ等(210,220,230)を介してインストールされて格納される。

【００８３】[ステップＡ５]開発されたアプリケーションプログラムＢをＡＳＰ基盤システムに適合されるため、提供された認証ＡＰＩ(303)及び課金ＡＰＩ(304)が、ベンダー端末(701)によって開発されたアプリケー

ションプログラムＢ(320)に組み込まれる。

【００８４】[ステップＡ６]認証ＡＰＩ(303)及び課金ＡＰＩ(304)が組み込まれたアプリケーションプログラムＢ(320)が、ベンダー端末(710)からアプリケーションサーバ１(300)にＬＡＮ等(210,220,230)を介してインストールされて格納される。

【００８５】[ステップＡ７]アプリケーションサーバ１(300)に格納されるアプリケーションプログラムＡ(310)及びアプリケーションプログラムＢ(320)のアプリケーション情報（機能概要、詳細説明、デモプログラムへのリンク、契約条件（料金表を含む））が、サーバ(500)に登録されて格納される。

【００８６】尚、アプリケーションサーバ１(300)に複数のアプリケーションプログラム(310,320,...)が格納される場合には、アプリケーションプログラム(310,320,...)のそれぞれに対するアプリケーション情報がサーバ(500)に登録されて格納される。

【００８７】また、図３はアプリケーションプログラムを利用するユーザの登録方法を説明するためのフローチャート図である。

【００８８】[ステップＢ１]

（アプリケーションプログラムを利用するユーザの登録方法）先ず、ユーザ端末(100)はサーバ(500)に接続する。

【００８９】[ステップＢ２]これに対し、サーバ(500)はアプリケーション情報をユーザ端末(100)に送信する。尚、サーバ(500)は、ユーザ端末(100)から入力される検索条件とアプリケーション情報に含まれる機能概要の一部とが一致するアプリケーション情報を検索し、検索条件に一致するアプリケーション情報をユーザ端末(100)に送信する手段を備える。

【００９０】[ステップＢ３]ユーザ端末(100)は検索条件を入力してアプリケーション情報を受信してユーザ端末(100)の表示手段（図示せず）に表示する。ここで、例えば、図４に示すアプリケーション情報表示画面が、ユーザ端末(100)の表示手段に表示される。図４を参照するに、表示されるアプリケーション情報表示画面は、アプリケーション名称（例えば、日英翻訳、百科事典）、機能概要の一覧である。ここで、ユーザは表示手段に表示される「詳細表示ボタン」、「デモ表示ボタン」、及び「料金表示ボタン」のそれぞれのボタンをマウスでクリックすることができる。「詳細表示ボタン」がクリックされると、アプリケーションプログラムの機能詳細説明画面（図示せず）が表示手段に表示される。また、「デモ表示ボタン」がクリックされると、アプリケーションプログラムのアプリケーションプログラムのデモプログラムが実行され、アプリケーションデモ画面（図示せず）が表示手段に表示される。加えて、「詳細表示ボタン」がクリックされると、料金表等の契約条件を表す契約条件画面（図示せず）が表示手段に表示される。

【００９１】[ステップＢ４]ユーザは、利用するアプリケーションプログラム（百科事典）を決めたら、当該アプリケーションプログラムの「ラジオボタン（選択）」をチェックして（図４中の◉）、「契約申し込みボタン」をクリックする。その後、ユーザ登録画面（図示せず）がユーザ端末(100)の表示手段に表示され、ユーザは登録に必要なユーザ情報（企業名、責任者名、住所、電話番号、決済方法、端末識別子等）を入力してサーバ(500)に送信する。ここで、端末識別子とは、ユーザ端末(100)のＩＰアドレス（IP address）若しくはフル・クォリファイド・ドメイン・ネイム（FQDN:Full Qualified Domain Name）である。例えば、ＩＰアドレスは192.168.01.001であり、FQDNはwww.○○○.co.jpである。

【００９２】[ステップＢ５]これに対し、サーバ(500)は必要なユーザ情報をユーザ端末(100)から受信し、利用に必要なユーザ情報（ユーザＩＤ（ユーザ識別）、パスワード）を発行する。サーバ(500)は、登録に必要なユーザ情報に前記利用に必要なユーザ情報を加えたユーザ情報（ユーザＩＤ、パスワード、企業名、責任者名、住所、電話番号、決済方法、端末識別子等）を登録して格納する。

【００９３】[ステップＢ６]サーバ(500)は、利用に必要な情報（ユーザＩＤ、パスワード）をユーザ端末(100)に送信して通知する。

【００９４】次に、[ステップＡ２]に係る認証ＡＰＩ(303)及び課金ＡＰＩ(304)をアプリケーションプログラムに組み込む方法につき、図６及び図７を参照して、説明する。図６は認証ＡＰＩ(303)をアプリケーションプログラムに組み込む方法を示すフローチャートである。このフローチャートで認証ＡＰＩ(303)をアプリケーションプログラムに組み込むを方法を説明する。まず組み込む前の一般的なアプリケーションプログラムの入力メッセージを処理するフローを説明する。

【００９５】[ステップＧ１]アプリケーションプログラムは、ユーザ端末(100)からメッセージ（会話要求）を受信する。

[ステップＧ２]アプリケーションプログラムは、受信したメッセージＩＤが処理ＩＤに等しいか比較する。受信したメッセージＩＤが処理１に等しければ該当する処理１を実行する（ステップＧ５へ）。受信したメッセージＩＤが処理１に等しくなければ次の処理ＩＤに等しいか比較する（ステップＧ３へ）。

[ステップＧ３]アプリケーションプログラムは、受信したメッセージＩＤが処理２に等しいか比較する。受信したメッセージＩＤが処理２に等しければ該当する処理２を実行する（ステップＧ６へ）。受信したメッセージＩＤが処理２に等しくなければ次の処理ＩＤに等しいか比較する（ステップＧ４へ）。

[ステップＧ４]アプリケーションプログラムは、受信し

たメッセージＩＤが処理ｎに等しいか比較する。受信したメッセージＩＤが処理ｎに等しければ該当する処理ｎを実行する（ステップＧ７へ）。受信したメッセージＩＤが処理ｎに等しくなければエラー処理を行う（ステップＧ８へ）。

[ステップＧ５]アプリケーションプログラムは、受信したメッセージを処理１して結果をＨＴＭＬ文書形式にしてユーザ端末(100)に送信する。

[ステップＧ６]アプリケーションプログラムは、受信したメッセージを処理２して結果をＨＴＭＬ文書形式にしてユーザ端末(100)に送信する。

[ステップＧ７]アプリケーションプログラムは、受信したメッセージを処理ｎして結果をＨＴＭＬ文書形式にしてユーザ端末(100)に送信する。

[ステップＧ８]アプリケーションプログラムは、受信したメッセージＩＤが最後の処理ＩＤと一致しなければエラーメッセージをユーザ端末(100)に送信する。

【００９６】次に認証ＡＰＩ(303)をアプリケーションプログラムに組み込む方法を説明する。ステップG1とステップＧ２の間に次に示すステップＨ１～Ｈ６を挿入する。尚、ステップＣ５において、動的メニュー生成システム(302)が暗号化された認証証明書をユーザ端末(100)に送信され、ステップＣ６において、暗号化された認証証明書がユーザ端末(100)からアプリケーションプログラムに送信されているとする。

[ステップＨ１]アプリケーションプログラムは、暗号化された認証証明書をクッキーとしてユーザ端末(100)から受信する。

[ステップＨ２]アプリケーションプログラムは、暗号化された認証証明書を認証ＡＰＩ(303)の引数にセットする。

[ステップＨ３]アプリケーションプログラムは、認証ＡＰＩ(303)を呼び出す。

[ステップＨ４]アプリケーションプログラムは、認証ＡＰＩ(303)から戻された値をチェックする。無効コードならステップＨ５へ行く。

[ステップＨ５]アプリケーションプログラムは、エラーメッセージとログイン画面を表示する。

[ステップＨ６]アプリケーションプログラムは、認証ＡＰＩ(303)から戻された新しい暗号化された認証証明書をクッキーとしてユーザ端末(100)に送信する。尚、ステップＨ１～ステップＨ６を実行させるためのプログラムを認証関数プログラムという。

【００９７】図７は課金ＡＰＩ(304)をアプリケーションプログラムに組み込む方法を示すフローチャートである。このフローチャートで認証ＡＰＩ(303)をアプリケーションプログラムに組み込むを方法を説明する。ステップＩ２が本来のアプリケーションの処理であり、図６のステップＨ５～Ｈ７が示す処理である。ステップＩ２（ステップＨ５～Ｈ７）の直前にステップＨ１を挿入

し、ステップＩ２の後ろにステップＩ３～ステップＩ５を挿入する。

[ステップＩ１]アプリケーションプログラムは、ステップＩ２の直前に処理開始時刻としてマシン時間を取得する。

[ステップＩ２]アプリケーションプログラムは、処理をするとともに、課金対象となる使用量を計測する。

[ステップＩ３]アプリケーションプログラムは、ステップＩ２の直後に処理終了時刻としてマシン時間を取得する。

[ステップＩ４]アプリケーションプログラムは、利用実績（ユーザ識別名、処理開始時刻、処理時間（=処理終了時刻−処理開始時刻）、ステップＩ２で採取した課金対象となる使用量等を含む。）を作成する。（図１０に示す利用実績(503-2)を参照。）

[ステップＩ５]アプリケーションプログラムは、課金ＡＰＩ(304)を呼び出す。尚、ステップＩ１～ステップＩ５を実行させるためのプログラムを課金関数プログラムという。

【００９８】次に、本発明の実施の形態のアプリケーション・サービスを提供する方法につき、図５、図８～図１０を参照して、説明する。

【００９９】[ステップＬ１]ユーザは、ユーザ端末(100)のＷｅｂブラウザ(101)を使用してコンピュータネットワーク(200)上に開設しているアプリケーションサーバ１(300)に接続する。

【０１００】[ステップＬ２]認証システム(301)が、Ｗｅｂサーバ(390)を介してログイン画面をユーザ端末(100)に送出する。

【０１０１】[ステップＬ３]ユーザは、ユーザ端末(100)の表示手段に表示されたログイン画面からログインＩＤ（ユーザ識別）とパスワードとを入力する。

【０１０２】[ステップＬ４]認証システム(301)は、ユーザ端末(100)から受信したログインＩＤとパスワードとユーザ端末(100)の端末識別子とをＷｅｂサーバ(390)から受け取る。ユーザ識別名、パスワード、端末識別子のすべてまたはあらかじめ決めてあるそれらの内のいくつかの組み合わせと、認証保持システム(501)からログインＩＤをキーにして受信した認証（ユーザＩＤ、パスワード、端末識別子を含む。図８．認証501-1参照。）を突き合わせて一致するか比較する。一致すれば正常、一致しなければ以上と判断する。

【０１０３】[ステップＬ５]認証システム(301)の認証結果が正常なら、認証したときの時刻（年月日時分秒）をマシン時間から取得する。次に、認証システム(301)は、ユーザＩＤと認証したときの時刻とを含む認証証明書を作成する。同時に、認証システム(301)は、認証証明書を暗号化し、暗号化した認証証明書を作成する。次に、認証システム(301)は、暗号化した認証証明書をキーとしてユーザ認証保持システム(501)に暗号化前の認

証証明書と暗号化した認証証明書と（図８．認証証明書501–2参照。）を保存する。そして認証システム(301)は動的メニュー生成システム(302)を起動し，ユーザＩＤと暗号化した認証証明書を引き渡す。ここで、認証証明書の暗号化には、第三者に解読不可能な一方向暗号アルゴリズムを使用する。たとえば要約関数MD5（RFC1321で定義されている）を使う。尚、MD5の出力結果はバイナリなのでネットワーク透過にするためBASE64でエンコードするしたものを暗号化した認証証明書とする。

【０１０４】[ステップＬ６]動的メニュー生成システム(302)は、ユーザＩＤをキーとしてメニュー保持システム(502)から当該ユーザが利用可能なアプリケーションプログラムのリスト（図９．アプリケーションプログラムリスト502–1参照。）と各アプリケーションプログラムが動作可能なサーバのリスト（図９．サーバリスト502–2参照。）を受信する。アプリケーションプログラムリスト(502–1)は、ユーザＩＤとアプリケーションプログラム名とを含む。また、サーバリスト(502–2)は、アプリケーションプログラム名とサーバ端末識別名とプログラムファイル名とを含む。

【０１０５】[ステップＬ７]動的メニュー生成システム(302)はクッキー化した，且つ暗号化した認証証明書と当該ユーザが利用可能なアプリケーションプログラムのリストをURI形式に合成し、メニューを生成する。動的メニュー生成システム(302)は生成したメニューをＷｅｂサーバ(390)を介してユーザ端末(100)に送出する。

【０１０６】[ステップＬ８]ユーザが、メニュー画面からアプリケーションＡ(310)を選択すると，Ｗｅｂブラウザ(101)はアプリケーションＡ(310)のURIを使って選択されたアプリケーションに接続する。このとき、ユーザ端末(100)は、ステップＬ７で受信した暗号化された認証証明書をクッキーとして送出する。

【０１０７】[ステップＬ９]アプリケーションＡ(310)は、Ｗｅｂサーバ(390)を介してクッキーとして暗号化された認証証明書を受信する。アプリケーションＡ(310)は、この暗号化された認証証明書を引数に格納して認証ＡＰＩ(303)を呼び出す。認証ＡＰＩ(303)は，認証システム(310)に暗号化された認証証明書を送出し、正当性確認を要求する。

【０１０８】[ステップＬ１０]認証システム(301)は、暗号化された認証証明書をキーとして認証保持システム(501)から認証証明書を取り出す。認証証明書に含まれる認証時刻が、現在のマシン時刻を基点としてあらかじめ決めておいた無通信タイムアウト時間より過去であれば、この暗号化された認証証明書は無効であると判定する。認証システム(301)は、保持システム(501)から取り出した認証証明書を再度暗号化したものが受信した暗号化された認証証明書と一致すれば有効、一致しなければ無効と判定する。認証システム(301)の認証結果が有効であれば、認証システム(301)は、認証証明書の認証時刻を現在のマシン時刻に変更して認証証明書を更新する。次に、認証システム(301)は、新しい認証証明書を暗号化する。認証システム(301)は、この暗号化した認証証明をキーとして新しい認証証明を認証保持システム(401)に登録し、古い認証証明は削除する。

【０１０９】[ステップＬ１１]認証システム(301)は、認証ＡＰＩ(303)を介してアプリケーションＡ(310)に認証結果を応答する。認証システム(301)の認証結果が有効なら、ユーザＩＤと新しい暗号化された認証証明書を返し、無効なら無効を意味するコードを返す。

【０１１０】[ステップＬ１２]アプリケーションＡ(310)は、認証システム(301)の認証結果が正常なら、ユーザ端末(100)とアプリケーションレベルの会話（メッセージ）を開始する。このとき，アプリケーションＡ(310)は、ユーザ端末(100)からWebサーバ(390)を介してを受信すると，必ず暗号化された認証証明書がクッキーとして送られてくるので、毎回ステップＬ９～ステップＬ１１の手順で正当性を確認する。アプリケーションＡ(310)は、ユーザ端末(100)にWebサーバ(390)を介してＨＴＭＬ文書を送信するときはステップＬ１１で返された新しい暗号化された認証証明書をクッキーとして送信する。

【０１１１】[ステップＬ１３]一連のアプリケーションレベルの会話（メッセージ：会話要求、会話返送）の途中、又はアプリケーションＡ(310)の終了時に、アプリケーションＡ(310)は利用実績(503–2)としてユーザ識別名、時刻、利用機能、各アプリケーションプログラムで決めた課金対象となるアプリケーション利用量等を課金ＡＰＩ(304)を介して課金システム(503)に送出する。課金システム(503)は，受信した利用実績を蓄積し、定期的に集計して，認証保持システム(501)に格納しているユーザ別アプリケーション別各アプリケーション利用量単価（図１０．単価マスタ503–1参照。）にユーザ別アプリケーション別アプリケーション利用量を積算し、利用料金の額を算出して請求書を作成する。

【０１１２】（実施の形態２）次に、本発明の実施の形態２のアプリケーション・サービスを提供する方法及びシステムにつき、図１１を参照して，説明する。図１１は本発明の実施の形態２のアプリケーション・サービスを提供するシステムの構成例を示すブロック図である。図５に示す本発明の実施の形態１のシステムは一のアプリケーションサーバ１(300)から構成されていたが、図１２に示す実施の形態２のシステムは複数のアプリケーションサーバ(300,400,...)から構成されてなる。

【０１１３】次に、本発明の実施の形態のアプリケーション・サービスを提供する方法につき、図１１を参照して，説明する。尚、ステップＬ１～ステップＬ１３に相当するものである。

【０１１４】[ステップＣ１:ログインステップ]アプリケーションプログラムに対する接続要求が、ユーザ端末

(100)から複数のアプリケーションサーバ(300,400,...)のうち何れか一のアプリケーションサーバ１(300)に送受信される。ここで、アプリケーションサーバ１(300)は、接続要求をＷｅｂサーバ(390)を介して認証システム(301)で受信している。これに対し、アプリケーションサーバ１(300)は、ログイン画面をＷｅｂサーバ(390)を介して認証システム(301)からユーザ端末(100)に送信する。一方、ユーザ端末(100)は、Ｗｅｂブラウザ(101)を介してログイン画面を受信すると、ログイン画面がユーザ端末(100)の表示手段に表示される。次に、ユーザは、ログインＩＤ（ユーザ識別）とパスワードとを入力してアプリケーションサーバ１(300)に送信する。これに対し、アプリケーションサーバ１(300)は、ログインＩＤ（ユーザ識別）とパスワードとをＷｅｂサーバ(390)を介して認証システム(301)に入力する。

【０１１５】尚、ログインＩＤ（ユーザ識別）とパスワードとのパケットがユーザ端末(100)からアプリケーションサーバ１(300)にＩＰプロトコルで送受信されるているので、パケットは常に送信元（ユーザ端末(100)のＩＰアドレス）と送信先（アプリケーションサーバ１(300)のＩＰプロトコル）とが格納されている。従って、アプリケーションサーバ１(300)は、ログインＩＤ（ユーザ識別）とパスワードとを受信するとともに、ユーザ端末(100)のＩＰアドレス（端末識別子）をパケットから抽出して認証システム(301)に入力することもできる。

【０１１６】[ステップＣ２:認証ステップ]次に、承認システム(301)は、入力されたログインＩＤ（ユーザ識別）及びパスワードと、認証保持システム(501)に予め格納されるユーザＩＤ（ユーザ識別）及びパスワードと、を比較する。この比較の結果として、入力されたログインＩＤ及びパスワードと、予め格納されるユーザＩＤ及びパスワードと、が一致する場合に、認証システム(301)は、ユーザ端末(100)のユーザを認証する。この時、認証システム(301)は、ユーザＩＤと認証した時刻とを含む認証証明書を作成して認証保持システム(501)に格納する。ここで、認証証明書が認証した時刻を含むことにより、認証証明書に有効期限を持たせることができる。また、認証証明書がユーザＩＤとともに認証した時刻を含むことにより、承認時に毎回異なるユニークな認証証明書を作成することができる。認証証明書が有効期限で時間で管理される、又は認証証明書が承認時に毎回異なるので、認証証明書の偽造を防止することができるという利点がある。加えて、認証システム(301)が暗号化した認証証明書を作成することにより、認証証明書の偽造を防止を強化することができる。

【０１１７】尚、認証システム(301)は、ＩＰアドレス（端末識別子）が入力される場合には、入力されたＩＰアドレス（端末識別子）と、予め格納される端末識別子と、を更に比較することができる。この比較の結果として、入力されたログインＩＤ、パスワード、及び端末識別子と、予め格納されるユーザＩＤ、パスワード、及び端末識別子と、が完全に一致する場合に、認証システム(301)は、ユーザ端末(100)のユーザを認証することもできる。これにより、認証システム(301)は、仮に入力されたログインＩＤ及びパスワードが偽造されたものであっても、正規なユーザのみを正確に認証することができる。

【０１１８】[ステップＣ３]次に、認証システム(301)は、動的メニュー生成システム(302)を起動し、認証証明書を動的メニュー生成システム(302)に入力する。

【０１１９】[ステップＣ４]これに対し、動的メニュー生成システム(302)は、入力された認証証明書のユーザＩＤをキーとして、予め当該ユーザが利用契約したアプリケーションプログラムリストをメニュー保持システム(702)から取得する。次に、動的メニュー生成システム(302)は、取得したアプリケーションプログラムリストのアプリケーションプログラム名をキーとしてサーバリストをメニュー保持システム(702)から取得する。ここで、取得したサーバリストが複数ある場合、動的メニュー生成システム(302)は、エラーステータス及び平均応答時間を複数のサーバリストから取得する。次に、動的メニュー生成システム(302)は、抽出したエラーステータスがサービス中であり、且つ抽出した平均応答時間が最小であるサーバリストを選択する。次に、動的メニュー生成システム(302)は、選択したサーバリストに係るプログラムファイル名及びサーバ端末識別名を取得する。次に、動的メニュー生成システム(302)は、取得したプログラムファイル名及びサーバ端末識別名に基づいてＵＲＩ(Universal Resource Identifire)情報を含むメニューを動的に作成する。尚、ＵＲＩは、IETFが公開している技術文書rfc2396で定義されている(http://www.ietf.org/rfc/rfc2396.txt)。また、ＵＲＩはＵＲＬ(Universal ResourceLocator)と同義である。

【０１２０】[ステップＣ５]次に、動的メニュー生成システム(302)は、メニューと認証証明書とをＷｅｂサーバ(390)を介してユーザ端末(100)に送信する。尚、認証証明書が暗号化されている場合には、暗号化された認証証明書のみがユーザ端末(100)に送信される。これに対し、ユーザ端末(100)は、メニューと認証証明書とをＷｅｂブラウザー(101)を介してＷｅｂサーバ(390)から受信する。ここで、Ｗｅｂブラウザー(101)は、認証証明書をクッキー(Cookie)として一次保持する。

【０１２１】[ステップＣ６]ユーザが、ユーザ端末(100)でアプリケーションプログラムＡ(310)を選択すると、メニューのＵＲＩに従って、ユーザ端末(100)は、アプリケーションサーバ１(300)に格納されるアプリケーションプログラムＡ(310)に対する会話要求(メッセージ)をＷｅｂブラウザー(101)を介してアプリケーションプログラムＡ(310)に送信する。この時、ユーザ端末(10

0)は、認証証明書をクッキー(Cookie)として送信する。これに対し、アプリケーションプログラムＡ(310)は、会話要求(メッセージ)と認証証明書とをＷｅｂサーバ(390)を介してアプリケーションプログラムＡ(310)で受信する。

【０１２２】[ステップＣ７]アプリケーションプログラムＡ(310)は、認証証明書をアプリケーションプログラムＡ(310)に組み込まれた認証ＡＰＩ(303)にセットする。ここで、ＡＰＩ(Application Program Interface)とは、認証システム(301)、課金システムプログラム(503)等の外部プログラムをアプリケーションプログラムＡ(310)内部から使用するための仕組みである。尚、具体的に言えば、ＡＰＩはクラス名、メソッド名、引数の並びで規定されている。即ち、アプリケーションプログラムＡ(310)は、認証証明書を認証ＡＰＩ(303)を介して認証システム(301)に入力する。ここで、認証ＡＰＩ(303)は、認証証明書を認証システム(301)に送り、認証証明書の正当性を確認している。

【０１２３】[ステップＣ８:再認証ステップ]これに対し、認証システム(301)は、認証ＡＰＩ(303)を介して入力された認証証明書と、予め認証した認証証明書と、を比較する。この比較の結果として、入力された認証証明書と、予め格納した認証証明書と、が一致する場合に、認証システム(301)は、ユーザ端末(100)のユーザを再度認証する。この時、認証システム(301)は、ユーザＩＤと再度認証した時刻とを含む認証証明書を作成して認証保持システム(501)に格納する。即ち、認証システム(301)は、認証証明書の認証時刻を更新して、認証保持システム(501)に格納する。次に、認証システム(301)は、再度認証された認証証明書を認証ＡＰＩ(303)を介してアプリケーションプログラムＡ(310)に出力する。

【０１２４】[ステップＣ９]次に、アプリケーションプログラムＡ(310)は、会話要求(メッセージ)に対する会話返答(メッセージ)を作成する。次に、アプリケーションプログラムＡ(310)は、作成した会話返答(メッセージ)と再度認証された認証証明書とをＷｅｂサーバ(390)を介してユーザ端末(100)に送信する。これに対し、ユーザ端末(100)は、作成された会話返答(メッセージ)と再度認証された認証証明書とをＷｅｂブラウザー(101)を介してＷｅｂサーバ(390)から受信する。ここで、再度認証された認証証明書は、Ｗｅｂサーバ(390)からＷｅｂブラウザー(101)にクッキー(Cookie)として送受信されている。この後、ユーザ端末１００とアプリケーションＡ(310)の間でメッセージの送受信があるたびに、認証証明書の認証及び更新を行う。これらにより、認証証明書の偽造を防止を更に強化することができる。

【０１２５】[ステップＣ１０]ユーザがアプリケーションプログラムＡ(310)の利用を終了すると、利用実績を課金ＡＰＩ(304)を介して課金システム(503)に出力する。

【０１２６】[ステップＣ１１]前記ステップＣ６のメニューに含まれるＵＲＩは、アプリケーションサーバ１(300)に格納されるアプリケーションプログラムＡ(310)を特定し得るものであった。即ち、ステップＣ４において、アプリケーションサーバ１(300)に格納されるアプリケーションプログラムＡ(310)の平均応答時間が最小であったので、ＵＲＩは、アプリケーションサーバ１(300)に格納されるアプリケーションプログラムＡ(310)を特定し得るものであった。ところが、アプリケーションサーバ２(400)に格納されるアプリケーションプログラムＡ(310)の平均応答時間が最小である場合、動的メニュー生成システム(302)が作成するメニューに係るＵＲＩは、アプリケーションサーバ２(400)に格納されるアプリケーションプログラムＡ(310)を特定し得るものになる。この場合、ユーザが、ユーザ端末(100)でアプリケーションプログラムＡ(310)を選択すると、メニューのＵＲＩに従って、ユーザ端末(100)は、アプリケーションサーバ２(400)に格納されるアプリケーションプログラムＡ(310)に対する会話要求(メッセージ)をＷｅｂブラウザー(101)を介してアプリケーションプログラムＡ(310)に送信する。この時、ユーザ端末(100)は、認証証明書をクッキー(Cookie)として送信する。これに対し、アプリケーションサーバ２(400)に格納されるアプリケーションプログラムＡ(310)は、会話要求(メッセージ)と認証証明書とをＷｅｂサーバ(390)を介してアプリケーションプログラムＡ(310)で受信する。

【０１２７】[ステップＣ１２]アプリケーションプログラムＡ(310)は、認証証明書を認証ＡＰＩ(303)を介して認証システム(301)に入力する。

【０１２８】[ステップＣ１３]これに対し、認証システム(301)は、認証ＡＰＩ(303)を介して入力された認証証明書と、予め認証した認証証明書と、を比較する。この比較の結果として、入力された認証証明書と、予め格納した認証証明書と、が一致する場合に、認証システム(301)は、ユーザ端末(100)のユーザを再度認証する。この時、認証システム(301)は、ユーザＩＤと再度認証した時刻とを含む認証証明書を作成して認証保持システム(501)に格納する。次に、認証システム(301)は、再度認証された認証証明書を認証ＡＰＩ(303)を介してアプリケーションプログラムＡ(310)に出力する。

【０１２９】このように、アプリケーションサーバ１(300)及びアプリケーションサーバ２(400)は、認証証明書をクッキーの仕組みで受信し、認証システム(301)を介して認証保持システム(501)に格納される認証証明書と比較して受信した認証証明書の正当性の確認を行うことができる。即ち、最初にユーザがログインしたアプリケーションサーバ１(300)に格納されるアプリケーションプログラムＡ(310)のほか、最初にログインしたアプリケーションサーバ１(300)と異なるアプリケーションサーバ２(400)に格納されるアプリケーションプログラム

Ａ(310)も同一のユーザを識別することができる。換言すれば、複数のアプリケーションサーバ間でユーザ管理の統一性を実現することができる。

【０１３０】[ステップＣ１４]ユーザがアプリケーションプログラムＡ(310)の利用を終了すると、利用実績を課金ＡＰＩ(304)を介して課金システム(503)に出力する。尚、アプリケーションプログラムが独自の利用実績出力機能を持っているときは、統一性のある利用実績のフォーマットに変換して課金システム(503)にファイル転送で送信する機能を有する。

【０１３１】次に、本発明の実施の形態のアプリケーション・サービスを提供する方法につき、図１２～図１４を参照して、簡単に説明する。図１２はユーザがアプリケーションサーバ１(300)に格納されるアプリケーションプログラムを利用した場合の流れ図である。

[ステップＤ１]ユーザ端末(100)は、アプリケーションサーバ１(300)に接続する。

[ステップＤ２]アプリケーションサーバ１(300)は、ユーザ端末(100)にログイン画面を送信する。

[ステップＤ３]ユーザ端末(100)は、アプリケーションサーバ１(300)にユーザＩＤとパスワードとを送信する。

[ステップＤ４]アプリケーションサーバ１(300)は、ユーザＩＤとパスワードと必要に応じてユーザ端末(100)の端末識別子をチェックし、正当と判断すれば、ユーザが利用契約したアプリケーションプログラムの一覧をメニューとして送信する。

[ステップＤ５]ユーザ端末(100)は、アプリケーションプログラムを選択するとＵＲＩに従ってアプリケーションサーバ１(300)に接続する。

[ステップＤ６]アプリケーションサーバ１(300)は、ＵＲＩに従って選択されたアプリケーションプログラムを起動する。

[ステップＤ７]ユーザは、ユーザ端末(100)でアプリケーションプログラムを利用し、終了する。

[ステップＤ８]ユーザが、アプリケーションサーバ１(300)でアプリケーションプログラムが終了すると、利用実績を蓄積する。

【０１３２】図１３はユーザがアプリケーションサーバ２(400)に格納されるアプリケーションプログラムを利用した場合の流れ図である。

[ステップＥ１～Ｅ４]ステップＤ１～Ｄ４と同じである。

[ステップＥ５]ユーザ端末(100)は、アプリケーションプログラムを選択するとＵＲＩに従ってアプリケーションサーバ２(400)に接続する。

[ステップＥ６]アプリケーションサーバ２(400)は、ＵＲＩに従って選択されたアプリケーションプログラムを起動する。

[ステップＥ７]ユーザは、ユーザ端末(100)でアプリケーションプログラムを利用し、終了する。

[ステップＥ８]ユーザが、アプリケーションサーバ２(400)でアプリケーションプログラムが終了すると、利用実績を蓄積する。

【０１３３】図１４は課金の流れ図である。

[ステップＦ１]サーバ(500)は、定期的に蓄積した利用実績をユーザ別アプリケーション別に集計する。

[ステップＦ２]サーバ(500)は、課金処理を実行して請求書を発行し、ユーザに電子メール、郵便等で送付する。

[ステップＦ３]ユーザは、利用料金をサーバ(500)に電子マネー、クレジットカード、口座振込等で支払う。

[ステップＦ４]サーバ(500)は、ベンダー別アプリケーション利用料金を集計する。

[ステップＦ５]サーバ(500)は、ベンダー端末(700)にアプリケーション利用料金を電子マネー、口座振込等で支払う。

【０１３４】

【発明の効果】本発明のアプリケーション・サービスを提供する方法及びシステムによれば、認証ＡＰＩ及び課金ＡＰＩを組み込んだアプリケーションプログラムと認証システムとをアプリケーションサーバに格納するだけで、ベンターはアプリケーション・サービスを提供することができる。即ち、ベンダーがアプリケーション・サービス・プロバイダの事業に容易に参入できるという効果がある。

【００１３５】また、ベンダーはユーザ認証及び課金処理プログラムを開発する必要がなくなるので、新たなアプリケーションプログラムを開発することができる。この場合、この新たなアプリケーションプログラムをアプリケーションサーバに格納するだけで、ベンターは複数のアプリケーション・サービスを提供することができる。即ち、ユーザが利用可能なアプリケーションプログラムの数若しくは種類が多くなるという効果がある。ここで、複数のアプリケーションプログラムがそれぞれ認証ＡＰＩを介して一の認証システムによって、ユーザを識別することができる。換言すれば、複数のアプリケーションプログラム間でユーザ管理の統一性を実現することができる。

【００１３６】更に、別のベンダーが別のアプリケーションプログラムを開発した場合、複数のアプリケーションプログラムと一の認証システムとをアプリケーションサーバに別のアプリケーションプログラムを格納するだけで、別のベンダーもアプリケーション・サービスを提供することができる。即ち、アプリケーションサーバは複数のベンダーのアプリケーションプログラムをユーザに有償で利用させることができる。ここで、複数のベンダーのアプリケーションプログラムがそれぞれ課金ＡＰＩを介して一の課金システムによって、利用実績を集計して利用金額を算出することができる。換言すれば、複

35

数のベンダーの間で課金管理の統一性を実現することができる。課金システムはユーザ別に利用金額を算出してユーザに支払いを求める一方、課金システムはベンダー別に利用金額を算出してベンダーに支払う。これにより、ユーザは利用代金の支払い等の事務手続が煩雑でないという効果がある。

【図面の簡単な説明】

【図１】　実施の形態１のアプリケーション・サービスを提供するための準備をするシステムの構成例を示すブロック図である。

【図２】　実施の形態１のアプリケーションプログラムを準備する方法を説明するためのフローチャート図である。

【図３】　実施の形態１のアプリケーションプログラムを利用するユーザの登録方法を説明するためのフローチャート図である。

【図４】　実施の形態１のアプリケーション情報表示画面である。

【図５】　実施の形態１のアプリケーション・サービスを提供するシステムの構成例を示すブロック図である。

【図６】　実施の形態１の認証ＡＰＩをアプリケーションプログラムに組み込む方法を示すフローチャートである。

【図７】　実施の形態１の課金ＡＰＩをアプリケーションプログラムに組み込む方法を示すフローチャートである。

【図８】　実施の形態１の認証保持システムが保持する例である。

【図９】　実施の形態１のメニュー保持システムが保持する例である。

【図１０】　実施の形態１の課金システムが保持する例である。

【図１１】　実施の形態２のアプリケーション・サービスを提供するシステムの構成例を示すブロック図である。

36

【図１２】　実施の形態２のアプリケーションサーバ１のアプリケーションプログラムを利用した場合の流れ図である。

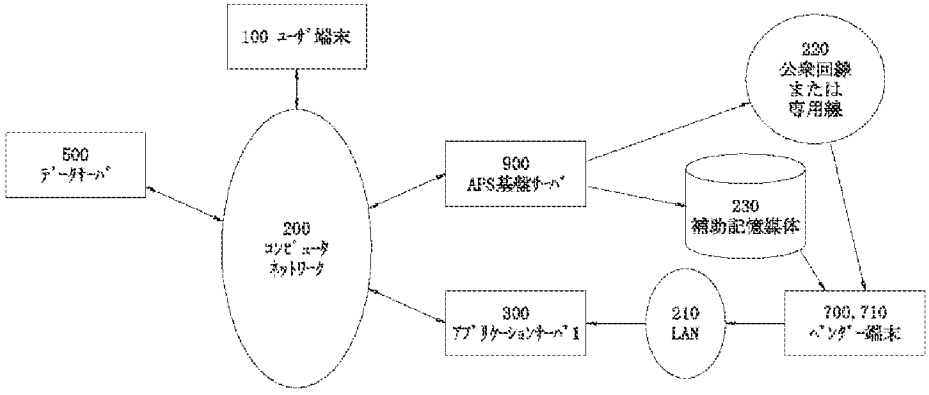
【図１３】　実施の形態２のアプリケーションサーバ２のアプリケーションプログラムを利用した場合の流れ図である。

【図１４】　実施の形態２の課金の流れ図である。

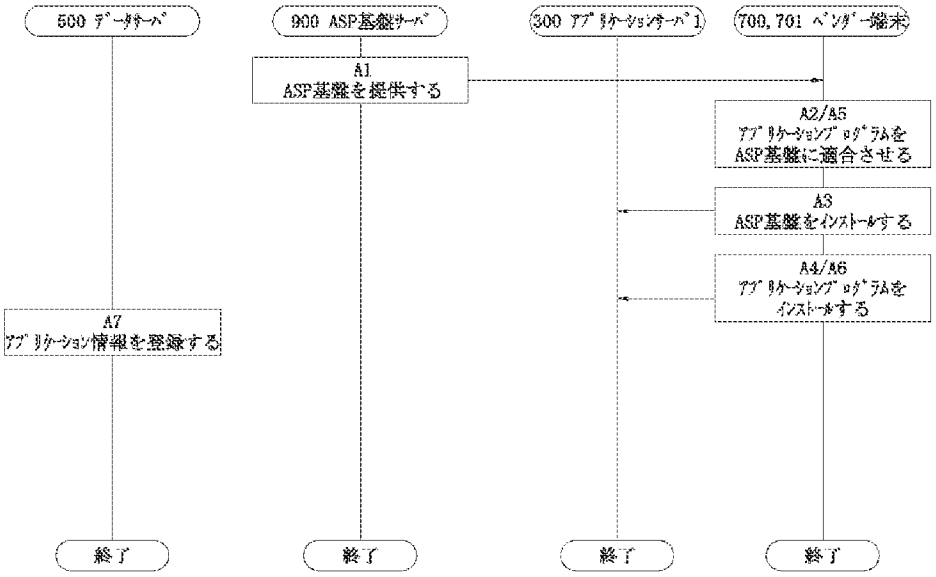
【符号の説明】

１００　　ユーザ端末
１０１　　Ｗｅｂブラウザ
２００　　コンピュータネットワーク
２１０　　ＬＡＮ
２２０　　公衆回線または専用線
２３０　　補助記憶媒体
３００、４００　　アプリケーションサーバ
３０１　　認証システム
３０２　　動的メニュー生成システム
３０３　　認証ＡＰＩ
３０４　　課金ＡＰＩ
３１０、３２０　　アプリケーションプログラム
３９０　　Ｗｅｂサーバ
５００　　サーバ
５０１　　認証保持システム
５０１－１　認証
５０１－２　認証証明書
５０２　　メニュー保持システム
５０２－１　アプリケーションプログラムリスト
５０２－２　サーバリスト
５０３　　課金システム
５０３－１　単価マスタ
５０３－２　利用実績
７００、７０１　　ベンダー端末
９００　　ＡＰＳ基板サーバ

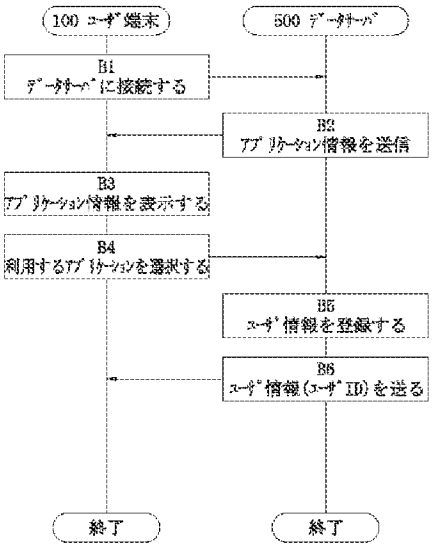
【図１】



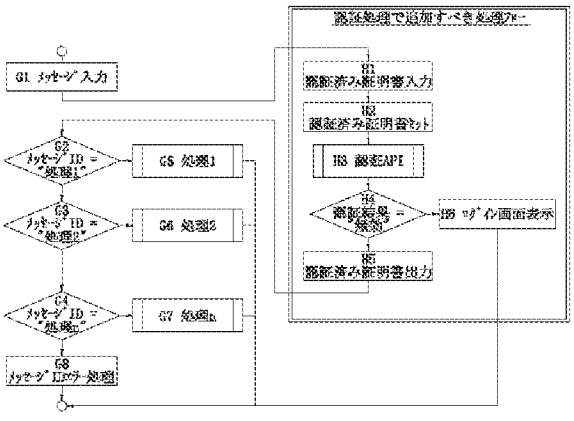
【図２】


500 データサーバ
900 ASP基盤サーバ
300 アプリケーションサーバ1
700,701 ベンダー端末
A1
ASP基盤を提供する
A2/A5
アプリケーションプログラムを
ASP基盤に適合させる
A3
ASP基盤をインストールする
A4/A6
アプリケーションプログラムを
インストールする
A7
アプリケーション情報を登録する
終了
終了
終了
終了

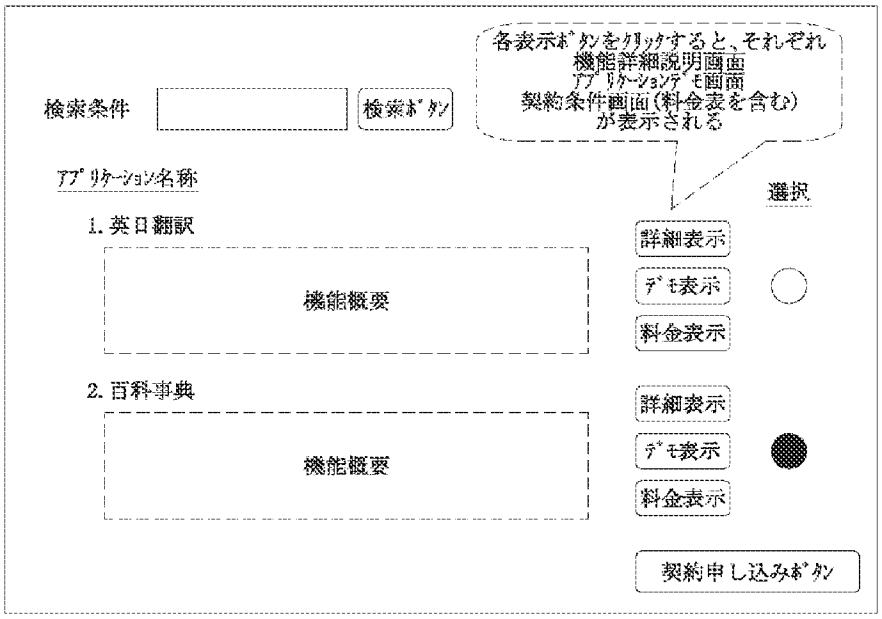
【図３】


100 ユーザ端末
500 データサーバ
B1
データサーバに接続する
B2
アプリケーション情報を送信
B3
アプリケーション情報を表示する
B4
利用するアプリケーションを選択する
B5
ユーザ情報を登録する
B6
ユーザ情報(ユーザID)を送る
終了
終了

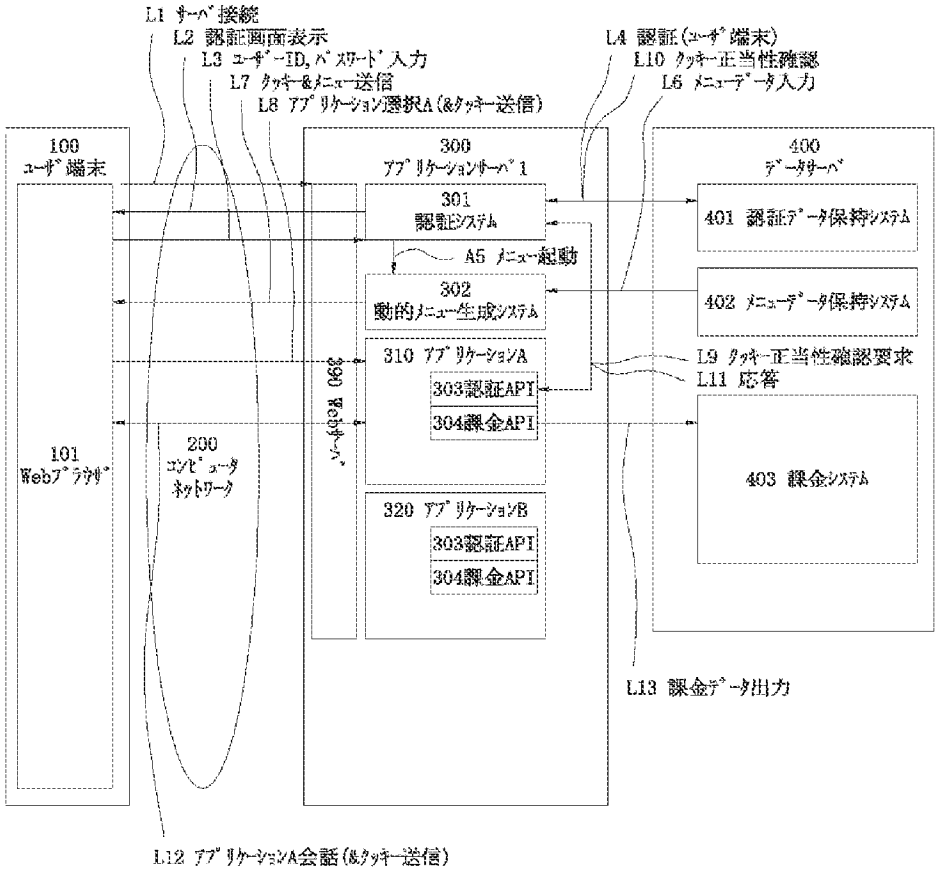
【図６】


認証処理で追加すべき処理フロー
G1 メッセージ入力
G2
メッセージID＝
"処理1"
G5 処理1
G3
メッセージID＝
"処理2"
G6 処理2
G4
メッセージID＝
"処理n"
G7 処理n
G8
メッセージIDエラー処理
H1
認証済み証明書入力
H2
認証済み証明書セット
H3 認証API
H4
認証結果＝
"成功"
H5 ログイン画面表示
H5
認証済み証明書出力

【図４】

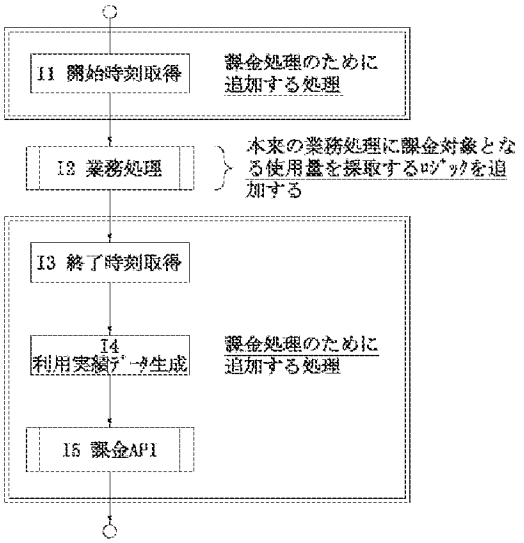

各表示ボタンをクリックすると、それぞれ
機能詳細説明画面
アプリケーションデモ画面
契約条件画面（料金表を含む）
が表示される
検索条件
検索ボタン
アプリケーション名称
選択
1. 英日翻訳
機能概要
詳細表示
デモ表示
料金表示
2. 百科事典
機能概要
詳細表示
デモ表示
料金表示
契約申し込みボタン

【図５】


L1 サーバ接続
L2 認証画面表示
L3 ユーザーID,パスワード入力
L7 クッキー&メニュー送信
L8 アプリケーション選択A(&クッキー送信)
L4 認証(ユーザ端末)
L10 クッキー正当性確認
L6 メニューデータ入力
100
ユーザ端末
101
Webブラウザ
200
コンピュータ
ネットワーク
300
アプリケーションサーバ1
301
認証システム
A5 メニュー起動
302
動的メニュー生成システム
310 アプリケーションA
303認証API
304課金API
320 アプリケーションB
303認証API
304課金API
390 Webサーバ
400
データサーバ
401 認証データ保持システム
402 メニューデータ保持システム
L9 クッキー正当性確認要求
L11 応答
403 課金システム
L13 課金データ出力
L12 アプリケーションA会話(&クッキー送信)

【図７】

(図6の処理1～nの内部の処理ﾌﾛｰ)



【図８】

認証ﾃﾞｰﾀ保持ｼｽﾃﾑ(501)で管理するﾃﾞｰﾀの一つの実施例

501-1 認証ﾃﾞｰﾀ

ﾕｰｻﾞID
ﾊﾟｽﾜｰﾄﾞ
端末識別子

501-2 認証証明書ﾃﾞｰﾀ

暗号化した認証証明書	
認証済み証明書	ﾕｰｻﾞID
	認証時刻(年月日時分秒)

【図９】

ﾒﾆｭｰﾃﾞｰﾀ保持ｼｽﾃﾑ(502)で管理するﾃﾞｰﾀの一つの実施例

502-1 ｱﾌﾟﾘｹｰｼｮﾝﾌﾟﾛｸﾞﾗﾑﾘｽﾄ

ﾕｰｻﾞID
ｱﾌﾟﾘｹｰｼｮﾝﾌﾟﾛｸﾞﾗﾑ名

502-2 ｻｰﾊﾞﾘｽﾄ

ｱﾌﾟﾘｹｰｼｮﾝﾌﾟﾛｸﾞﾗﾑ名
ｻｰﾊﾞ端末識別名
ﾌﾟﾛｸﾞﾗﾑﾌｧｲﾙ名

【図１０】

課金ｼｽﾃﾑ(503)で管理するﾃﾞｰﾀの一つの実施例

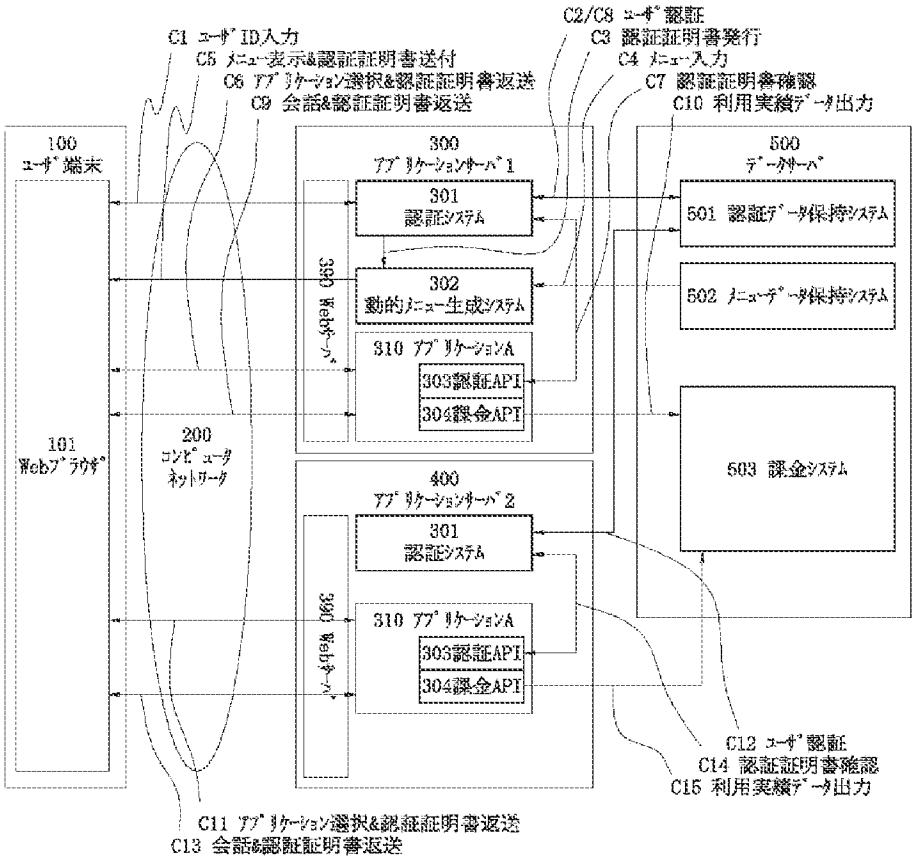
503-1 単価ﾏｽﾀ

ﾕｰｻﾞID
ｱﾌﾟﾘｹｰｼｮﾝﾌﾟﾛｸﾞﾗﾑ名
基本料金
課金項目1の単価
課金項目2の単価
・・・

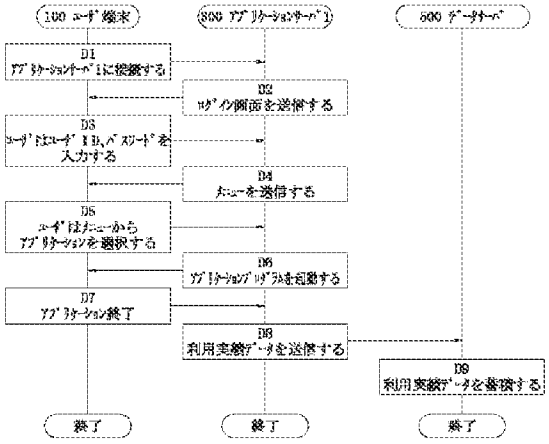
503-2 利用実績ﾃﾞｰﾀ

ﾕｰｻﾞID
ｱﾌﾟﾘｹｰｼｮﾝﾌﾟﾛｸﾞﾗﾑ名
処理開始時刻
処理時間
課金項目1
課金項目2
・・・

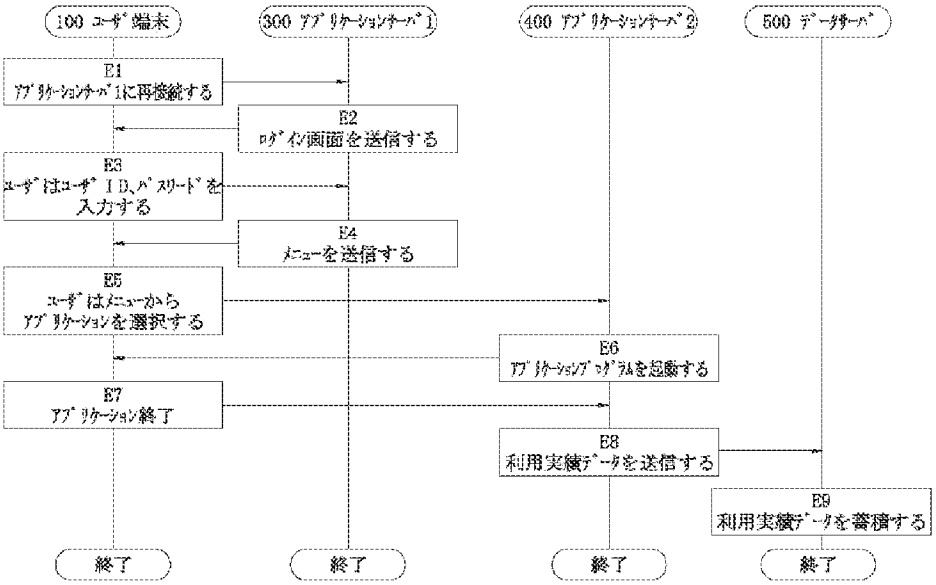
【図１１】


C1 ユーザID入力
C5 メニュー表示&認証証明書送付
C6 アプリケーション選択&認証証明書返送
C9 会話&認証証明書返送
C2/C8 ユーザ認証
C3 認証証明書発行
C4 メニュー入力
C7 認証証明書確認
C10 利用実績データ出力
100
ユーザ端末
101
Webブラウザ
200
コンピュータ
ネットワーク
300
アプリケーションサーバ1
301
認証システム
302
動的メニュー生成システム
310 アプリケーションA
303認証API
304課金API
390 Webサーバ
400
アプリケーションサーバ2
301
認証システム
310 アプリケーションA
303認証API
304課金API
390 Webサーバ
500
データサーバ
501 認証データ保持システム
502 メニューデータ保持システム
503 課金システム
C12 ユーザ認証
C14 認証証明書確認
C15 利用実績データ出力
C11 アプリケーション選択&認証証明書返送
C13 会話&認証証明書返送

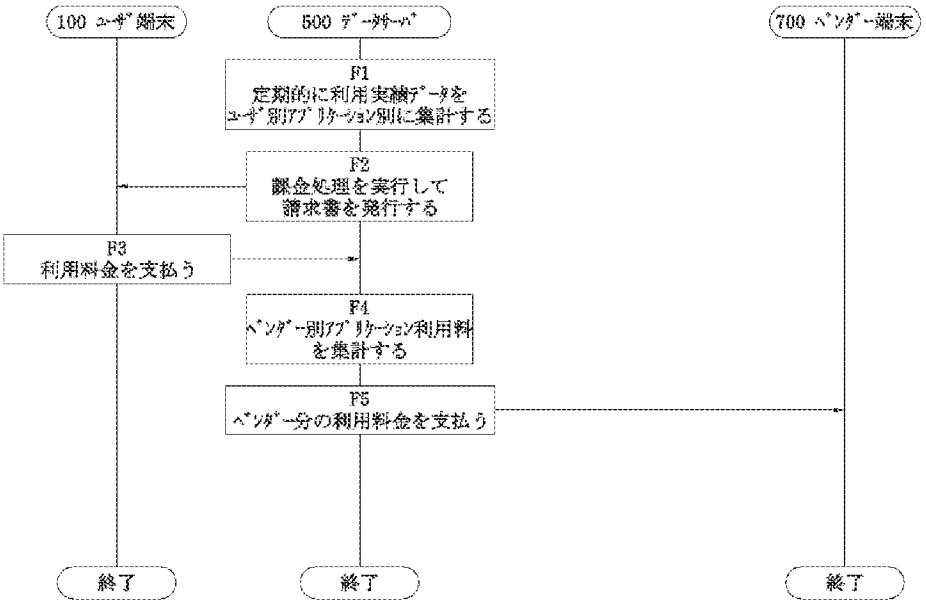
【図１２】


100 ユーザ端末
300 アプリケーションサーバ1
500 データサーバ
D1
アプリケーションサーバ1に接続する
D2
ログイン画面を送信する
D3
ユーザはユーザID,パスワードを入力する
D4
メニューを送信する
D5
ユーザはメニューからアプリケーションを選択する
D6
アプリケーションプログラムを起動する
D7
アプリケーション終了
D8
利用実績データを送信する
D9
利用実績データを蓄積する
終了
終了
終了

【図１３】


100 ユーザ端末
300 アプリケーションサーバ1
400 アプリケーションサーバ2
500 データサーバ
E1
アプリケーションサーバ1に再接続する
E2
ログイン画面を送信する
E3
ユーザはユーザID、パスワードを入力する
E4
メニューを送信する
E5
ユーザはメニューからアプリケーションを選択する
E6
アプリケーションプログラムを起動する
E7
アプリケーション終了
E8
利用実績データを送信する
E9
利用実績データを蓄積する
終了
終了
終了
終了

【図１４】


100 ユーザ端末
500 データサーバ
700 ベンダー端末
F1
定期的に利用実績データをユーザ別アプリケーション別に集計する
F2
課金処理を実行して請求書を発行する
F3
利用料金を支払う
F4
ベンダー別アプリケーション利用料を集計する
F5
ベンダー分の利用料金を支払う
終了
終了
終了